

MEMORY SYSTEM

Patent number: WO0049488
Publication date: 2000-08-24
Inventor: SINCLAIR ALAN WELSH (US);
OUSPENSKAIA NATALIA VICTOROVNA
(RU); TAYLOR RICHARD MICHAEL (GB);
GOROBETS SERGEY ANATOLIEVICH
(GB)
Applicant: MEMORY CORP PLC (GB); SINCLAIR
ALAN WELSH (US); OUSPENSKAIA
NATALIA VICTOROVNA (RU); TAYLOR
RICHARD MICHAEL (GB); GOROBETS
SERGEY ANATOLIEVICH (GB)
Classification:
- international: **G06F3/06; G06F3/06; (IPC1-7): G06F3/06**
- european: **G06F3/06E**
Application number: WO2000GB00550 20000217
Priority number(s): GB19990003490 19990217

Also published as:

EP1157328 (A1)
US6725321 (B1)
EP1157328 (B1)
DE60019903T (T2)

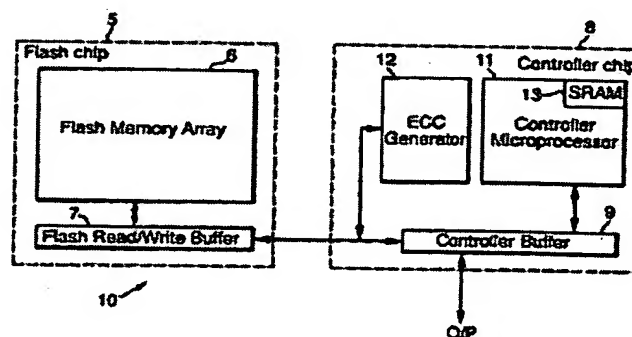
Cited documents:

GB2291991
WO9737296
EP0712067
EP0522780
WO9420906
more >>

Report a data error here

Abstract of WO0049488

A memory system (10) having a solid state memory (6) comprising non-volatile individually addressable memory sectors (1) arranged in erasable blocks, and a controller (8) for writing to reading from the sectors, and for sorting the blocks into "erased" and "not erased" blocks. The controller performs logical to physical address translation, and includes a Write Pointer (WP) for pointing to the physical sector address to which data is to be written from a host processor. A Sector Allocation Table (SAT) of logical addresses with respective physical addresses is stored in the memory, and the controller updates the SAT less frequently than sectors are written to with data from the host processor. The memory may be in a single chip, or in a plurality of chips. A novel system for arranging data in the individual sectors (1) is also claimed.



BEST AVAILABLE COPY

Data supplied from the **esp@cenet** database - Worldwide

(19) 日本国特許庁 (J P)

(12) 公表特許公報 (A)

(11) 特許出願公表番号

特表2002-537596

(P2002-537596A)

(43) 公表日 平成14年11月5日 (2002.11.5)

(51) Int.Cl. ⁷	識別記号	F I	テマコード* (参考)
G 0 6 F 12/02	5 7 0	G 0 6 F 12/02	5 7 0 A 5 B 0 1 8
3/08		3/08	H 5 B 0 6 0
12/00	5 9 7	12/00	5 9 7 U 5 B 0 6 5
12/16	3 1 0	12/16	3 1 0 P

審査請求 未請求 予備審査請求 有 (全106頁)

(21) 出願番号 特願2000-600167(P2000-600167)
(86) (22) 出願日 平成12年2月17日 (2000.2.17)
(85) 翻訳文提出日 平成13年8月17日 (2001.8.17)
(86) 国際出願番号 P C T / G B 0 0 / 0 0 5 5 0
(87) 国際公開番号 W O 0 0 / 4 9 4 8 8
(87) 国際公開日 平成12年8月24日 (2000.8.24)
(31) 優先権主張番号 9 9 0 3 4 9 0 . 2
(32) 優先日 平成11年2月17日 (1999.2.17)
(33) 優先権主張国 イギリス (GB)
(81) 指定国 EP (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, I T, LU, MC, NL, PT, SE), GB, J P, K R, S G, U S

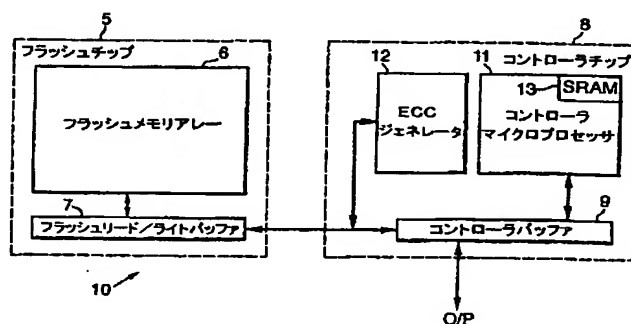
(71) 出願人 メンクエスト インコーポレイテッド
アメリカ合衆国 19801 ニューキャッスル郡 デラウェア州 ウィルミントン オレンジストリート 1209
(72) 発明者 シンクレア アラン ウェルシュ
アメリカ合衆国 カリフォルニア 92014、デルマー、マンゴドライブ ディー号 14059
(74) 代理人 弁理士 朝倉 正幸

最終頁に続く

(54) 【発明の名称】 メモリシステム

(57) 【要約】

メモリシステム (10) は、消去可能なブロック内に配置され且つ個々にアクセス可能な不揮発性メモリセクタ (1) と、セクタへの書き込み及びセクタからの読み込みを行ない且つブロックを消去ブロックおよび非消去ブロックにソートするためのコントローラ (8) とを備えた半導体メモリ (6) を有している。コントローラは、論理アドレスを物理アドレスに変換するとともに、ホストプロセッサからデータが書き込まれる物理セクタアドレスを指し示すライトポインタ (WP) を有している。各物理アドレスを有する論理アドレスのセクタ割り当てテーブル (S A T) はメモリ内に記憶される。コントローラは、ホストプロセッサからのデータがセクタに書き込まれる頻度よりも少ない頻度で S A T を更新する。メモリは、単一のチップであっても良く、あるいは、複数のチップであっても良い。また、個々のセクタ (1) にデータを配置する新規なシステムも請求の範囲に記載されている。



【特許請求の範囲】

【請求項1】 個別にアドレス可能で、かつ、セクタの消去可能ブロック内に設けられた不揮発メモリセクタを有し、前記セクタは、各々、メモリ内の自身の物理的位置を定義する物理アドレスを含む半導体メモリと；

データ構造のメモリへの書き込みとメモリからの読出しを行い、かつ、消去用として処理されるブロックと非消去用として処理されるブロックとにセクタのブロックを分別するためのコントローラとを具備するホストプロセッサに接続するためのメモリシステムであって、

前記コントローラは、

ホストプロセッサから受け取った論理アドレスをメモリ内の前記メモリセクタの物理アドレスに翻訳する手段と；

ホストプロセッサからのデータが書き込まれるセクタの物理アドレスを指示するライトポインタ（WP）とを備え、該ライトポインタ（WP）は、消去用として処理される任意のブロックにおけるメモリセクタの物理アドレスをくまなく所定の順番で移動するように、また、ブロックが満杯の場合は、別の消去ブロックに移動するようにコントローラによって制御され、

ホストプロセッサからセクタ書き込みコマンドを受けると、前記ライトポインタ（WP）がその時点で指示する物理アドレスを論理アドレスに割当てることによって、コントローラはホストプロセッサから受け取った論理アドレスをデータが書き込みされる物理アドレスに翻訳し、かつ、コントローラによって既に割当てられている各々の物理アドレスで論理アドレスのセクタ割当てテーブル（SAT）をコンパイルして、ホストプロセッサからのデータがメモリセクタに書き込まれる頻度より少ない頻度でSATを更新するよう構成されたシステム。

【請求項2】 前記ライトポインタ（WP）は、消去用に処理されるブロック中を所定の順序で移動するようコントローラによって制御されている請求項1に記載のメモリシステム。

【請求項3】 SATの物理セクタアドレスは、論理セクタアドレス（LSA）によって命令され、N番目のSATエントリーは、論理アドレスNを有するデータが書き込まれたセクタの物理アドレスを備える請求項1または2に記載

のメモリシステム。

【請求項4】 前記コントローラは、ホストプロセッサからセクタ読出しコマンドを受け取ると、以前、前記論理セクタアドレスに割当てた物理セクタアドレスを得るために、SAT内において、ホストプロセッサから受け取った論理セクタアドレス(LSA)を検索するように構成された請求項3に記載のメモリシステム。

【請求項5】 SATが、半導体メモリのメモリセクタの前記ブロックのうち、少なくとも1つに記憶されている前記請求項の何れかに記載のメモリシステム。

【請求項6】 コントローラは、ブロック全体におけるSATを再書き込みすることでSATを更新するよう構成されている請求項5に記載のメモリシステム。

【請求項7】 セクタブロックのうち、少なくとも1個のブロック(ASB)がSATブロックの個別セクタの修正バージョンを含んでいる請求項5または6に記載のメモリシステム。

【請求項8】 前記ASBブロックのセクタが各々、それが更新するSATブロックのセクタの物理アドレスを含む請求項7に記載のメモリシステム。

【請求項9】 前記ASBブロックの全てのセクタがSATセクタの修正バージョンを書き込まれるとき、SATブロックはそれぞれASBブロック内の全ての修正バージョンを含むように再書き込みされ、かつ、ASBブロックが消去される請求項7または8に記載のメモリシステム。

【請求項10】 コントローラがライトポインタ(WP)を制御して、各ブロックが書き込みデータで満杯になると、物理アドレスの番号昇順で、消去ブロック中を連続的に移動するように構成されている前記請求項の何れかに記載のメモリシステム。

【請求項11】 物理アドレス順で最高位のブロックセクタがひとたびデータで満杯になると、コントローラによる消去処理途中の全てのブロック中、番号的に最下位の物理アドレスを有するセクタのブロックを取り囲むようにWPがコントローラによって制御されるという意味から、ライトポインタ(WP)の制御

が循環式である請求項10に記載のメモリシステム。

【請求項12】 コントローラがライトポインタ(WP)を制御して、消去ブロック中を物理アドレス順で非連続的に移動するように構成されている請求項1～9の何れかに記載のメモリシステム。

【請求項13】 前記メモリセクタ(1)はそれぞれ、物理的にデータエリア(300)とスペアエリア(302)とに分割され、コントローラは、セクタのデータエリア(300)の始めからオフセットされるセクタ位置において、ヘッダデータおよびエラー修正コードデータ(ECC)を含むオーバーヘッドデータ(OD)を書き込みし、かつ、ホストプロセッサから受け取ったユーザデータをオーバーヘッドデータ(OD)の何れか一方の側のセクタにおける残りスペースに書き込みする構成されている請求項1～12の何れかに記載のメモリシステム。

【請求項14】 前記オーバーヘッドデータ(OD)は、セクタに書き込みされるユーザデータのうち少なくとも1ビットによって決定される総計によってオフセットされる請求項13に記載のメモリシステム。

【請求項15】 前記セクタブロックの各々におけるメモリセクタは、ユニットとしてまとめて消去できる前記請求項の何れかに記載のメモリシステム。

【請求項16】 前期セクタブロックの各々におけるメモリセクタは、個別に消去することもできる請求項15に記載のメモリシステム。

【請求項17】 コントローラがメモリの消去操作を制御してメモリセクタのブロック全体のみを消去し、その中のメモリセクタが全て消去セクタである場合は、セクタのブロックがコントローラによって消去ブロックとして処理される前記請求項の何れかに記載のメモリシステム。

【請求項18】 1つのブロックが1個または2個以上の不良セクタを含んでいる場合、コントローラがそのブロック全体を不良と定義し、非消去ブロックとして処理し、それによりデータがそのブロックに書き込みされない請求項17に記載のメモリシステム。

【請求項19】 1つのブロックが1個または2個以上の不良セクタを含んでいる場合、コントローラがそのブロックを消去ブロックとして処理し、それに

よりコントローラは該ブロック内の良好セクタをデータ保存用に使用し、メモリシステムは不良セクタを識別するテーブルを含み、コントローラは、次にライトポインタ (WP) が移動するセクタアドレスが不良セクタかどうかをチェックし、アドレスが不良セクタの場合は、その不良セクタをスキップして、次に書き込みが行われるセクタのアドレスに所定の順序で移動させる請求項 16 に記載のメモリシステム。

【請求項 20】 セクタブロックは各々、メモリ内での自身の物理的位置を定義する物理ブロックアドレスを有し、前記メモリセクタの各々の物理アドレスは、それが位置するブロックの物理ブロックアドレスを含み、コントローラは、消去処理されるセクタブロックの少なくとも何個かの物理ブロックのアドレスをブロック内を WP が移動する順番に並べたリストをコンパイルし、該リストを利用して、次に書き込みが行われるセクタブロックを素早く識別し、コントローラが前記リストを保存するための一時的記憶手段をさらに含んでなる前記請求項の何れかに記載のメモリシステム。

【請求項 21】 コントローラが以前に別のセクタに書き込みされた旧データを伝えるセクタ書き込みコマンドをホストプロセッサから受け取ると、その時点の旧データを有するセクタのアドレスをメモリシステムの一時的記憶域に保存する前記請求項の何れかに記載のメモリシステム。

【請求項 22】 コントローラがホストプロセッサからユーザによるセクタ除去コマンドを受け取ると、除去すべきセクタを廃棄としてマークし、かつ、該セクタのアドレスを前記一時的記憶手段に保存するよう構成されている請求項 21 に記載のメモリシステム。

【請求項 23】 コントローラは、いかなる場合も、ライトポインタ (WP) によって書き込みされた旧データを有する 1 個または 2 個以上のセクタを含むブロック (以下、これをカレントオブソリートブロック (COB) と称す) を予め決められた数しか容認せず、その結果、前記 COB 内の全てのセクタが旧データを含むと、前記 COB は即座に消去される請求項 21 または 22 に記載のメモリシステム。

【請求項 24】 COB 以外のブロックにあるセクタが旧データを含む場合

、コントローラは、前記COB内の有効（非廃棄）セクタにある任意のデータを別のブロックに再配置し、その後前記COBを消去し；COB以外のブロックにある前記セクタを廃棄としてマークし；前記別のブロックを新しいCOBとして指定する請求項23に記載のメモリシステム。

【請求項25】 前記の予め決められたCOBsの数が1個である請求項23または24に記載のメモリシステム。

【請求項26】 コントローラが前記有効データを再配置するブロックは、ライトポインタ（WP）がその時点で位置するブロックである請求項24に記載のメモリシステム。

【請求項27】 前記有効データが再配置されるセクタの物理アドレスを指し示すために別のライトポインタ（以下、リロケーションポインタ（RP）と称す）を備え、リロケーションポインタ（RP）は常にライトポインタ（WP）とは異なるセクタブロックに位置する請求項24に記載のメモリシステム。

【請求項28】 ホストプロセッサからのシステムデータが書き込まれるセクタの物理アドレスを指し示すためにさらに別のライトポインタ（以下、システムライトポインタ（SWP）と称す）を備え、システムライトポインタ（SWP）は常にライトポインタ（WP）とは異なるブロックに位置する請求項27に記載のメモリシステム。

【請求項29】 コントローラは1つまたは2つ以上の廃棄セクタを含む少なくとも2個のブロックが任意のときに存在することを容認し、前記ブロックの一方は前記COBであり、他方は1つまたは2つ以上の旧システムデータセクタを含むカレント・オブソリート・システム・ブロック（COSB）であり、COSBの消去を容認するために任意のシステムデータセクタを再配置することが必要になった場合、再配置されたシステムデータはシステムライトポインタ（SWP）がその時点で指示するアドレスに送られる請求項28に記載のメモリシステム。

【請求項30】 有効システムデータが再配置されるセクタの物理アドレスを指し示すために、さらに別のライトポインタ（以下、システムリロケーションポインタ（SRP）と称す）を備え、システムリロケーションポインタ（SRP）は常にライトポインタ（WP）及びシステムライトポインタ（SWP）

）とは異なるセクタブロックに位置する請求項 28 に記載のメモリシステム。

【請求項 31】 コントローラによる COB の消去が必要なときに COB が前記ライトポインタ (WP、RP、SWP、SRP) のうち 1 つを含んでいる場合、旧データは既に別のブロックに作成されているので、コントローラは新しい COB の作成を実行する一方、旧 COB の消去を保留し (以下、保留オブソリートブロック (POB) と称す)、これは POB 内の全消去セクタが充填され、かつ、コントローラの定義に従って前記ポインタが次に使用される消去ブロックに移動するまで保留され、このとき POB 内の任意の有効 (非廃棄) データはコントローラによって再配置され、POB が消去される請求項 28 ～ 30 の何れかに記載のメモリシステム。

【請求項 32】 SAT が最後に更新されてから、RP (以下、リロケーションセクタリストまたは RSL と称す) と、SWP (以下、ライトシステムセクタリストまたは WSSL と称す) と、SRP (以下、システムリロケーションセクタリストまたは SSSL と称す) とによってリロケーションデータが書き込まれたメモリ内のセクタに対応する論理セクタアドレスのそれぞれのリストをメモリシステムの一時的記憶域に保存するように、また、RP と、SWP と、SRP (以下、リロケーションブロックリスト (RBL) と称す) と、ライトシステムブロックリスト (WSBL) と、システムリロケーションブロックリスト (SRBL) とによって使用されたブロック順序に対応するリストを前記一時的記憶域に保存するようにコントローラが構成される請求項 29 に記載のメモリシステム。

【請求項 33】 ホストプロセッサからのメモリへのデータ構造の書き込みに加えて、コントローラは、制御情報として示されるメモリデータの生成および書き込みを行い、また、ホストプロセッサから受け取ったデータ構造が書き込まれるメモリセクタブロックのうち、1 個または 2 個以上の異なるブロック (制御ブロックまたは C Bs) にそうした制御情報を書き込むように構成された前記請求項の何れかに記載のメモリシステム。

【請求項 34】 コントローラは、全 SAT ブロックのブロックアドレスリストを少なくとも 1 個の前記制御ブロックに記憶する請求項 33 に記載のメモリ

システム。

【請求項35】 コントローラは前記1個または2個以上の制御ブロックのブロックアドレスをメモリの専用ブロック（ブートブロックまたはBB）に記憶し、この専用ブロックはいかなる不良セクタも含まないメモリセクタの第1ブロックである請求項33または34に記載のメモリシステム。

【請求項36】 全SATブロックアドレスの前記リストは複数のリスト部（テーブルブロックリストまたはTBLS）形式であり、かつ、各リスト部が論理的に連続する一群のSATブロックのブロックアドレスを含む請求項7、8または9に従属する請求項34に記載のメモリシステム。

【請求項37】 コントローラは、SATが最後に更新されてからライトポインタ（WP）によって書き込みされたデータ構造用の論理セクタアドレスのリスト（ライトセクタリストまたはWSL）をメモリシステムの一時的記憶域に保存する請求項7～9の何れかに記載のメモリシステム。

【請求項38】 コントローラはSATが最後に更新されてからデータを書き込むためにライトポインタ（WP）によって使用されたブロックの順番を前記一時的記憶域に保存し、この順番は更新されたセクタのアドレスがWSLに保持されるブロックのブロックアドレスリスト（ライトブロックリストまたはWBL）の形式で記憶される請求項37に記載のメモリシステム。

【請求項39】 WSLは予め決められた大きさで、ひとたびWSLが一杯になると少なくとも1個のSATブロックまたはASBブロックが更新され、WSLおよびWBLが空になる請求項38に記載のメモリシステム。

【請求項40】 コントローラは、開始物理セクタアドレスと、SATブロックまたはASBブロックが最後に更新されてからデータが書き込みされたセクタを有するブロック間のリンクとを半導体メモリの前記制御ブロックに記憶する請求項38に記載のメモリシステム。

【請求項41】 前記セクタは各々メモリの単一“ページ”、すなわち、メモリセクタの前記ブロックにあるメモリセルの一系列から成る前記請求項の何れかに記載のメモリシステム。

【請求項42】 コントローラは、均一にサイズ設定されたデータセグメン

ト内のメモリセクタに対してデータを書き込み、また、前記メモリセクタからデータを読み込む前記請求項の何れかに記載のメモリシステム。

【請求項43】 全てのメモリセクタが同一サイズであり、前記データセグメントは各々、前記メモリセクタと均等なサイズである請求項42に記載のメモリシステム。

【請求項44】 コントローラがSATから最も新しくアクセスしたSATを含む一群の連続SATエンタリーを記憶させる一時的キャッシュメモリをさらに含む前記請求項の何れかに記載のメモリシステム。

【請求項45】 コントローラは、SATセクタの書き込み操作が実行されるたびに更新される全てのASBブロックとこれらのASBブロックが連想させるSATブロックの物理アドレスのリスト(ASBL)を前記一時的キャッシュメモリ内に作成する請求項8に従属する請求項43に記載のメモリシステム。

【請求項46】 半導体メモリは単一のメモリチップとして形成された単一のメモリアレイを備える前記請求項の何れかに記載のメモリシステム。

【請求項47】 半導体メモリは複数のメモリチップで形成されたメモリアレイを備える請求項1～45の何れかに記載のメモリシステム。

【請求項48】 半導体メモリは複数のメモリチップで形成された複数のメモリアレイを含み、コントローラは複数のメモリチップにあるメモリセクタから多重仮想ブロックを形成し、前記仮想ブロックはそれぞれ、前記メモリチップの各々のメモリセクタのうち1個の消去可能ブロックを含み、コントローラが前記仮想ブロックを消去用として処理するものと、非消去用として処理するものとに分別する請求項1～45の何れかに記載のメモリシステム。

【請求項49】 コントローラは、消去処理される仮想ブロックのリストをコンパイルし、これをメモリシステムの一時的記憶域に保存し、また、各々の連続セクタ書き込み操作のために1つのチップから別のチップに移動するようにライトポインタ(WP)を制御し、この移動は、仮想ブロックの1つの消去可能ブロックにある1個のセクタから始まり、仮想ブロックの各消去可能ブロックにあるセクタの書き込みが完了するまで、仮想ブロック内にあるその他の各消去可能ブロックのセクタを連続して移動し、次に最初のセクタが書き込みされたチップ

に戻って、仮想ブロックがデータで満杯になるまで同様の方法で仮想ブロック内の各消去可能ブロックにある別のセクタへの書き込みを続け、その後、前記消去処理される仮想ブロックリストに載っている次の仮想ブロックにライトポインタ(WP)を移動し、この仮想ブロックを同様の手法で書き込みする請求項48に記載のメモリシステム。

【請求項50】 n回の連続セクタ書き込み操作のたびにコントローラがホストプロセッサから受け取った多重セクタ書き込みコマンドを実行し、nがメモリシステムの半導体メモリチップの数より少ないかまたは等しい場合、コントローラがチップの各nにおける1個のセクタに連続して同時に書き込みする請求項49に記載のメモリシステム。

【請求項51】 コントローラは、仮想ブロックにある全ての消去可能ブロックを同時に消去することで任意の前記仮想ブロックの消去を実行する請求項49または50に記載のメモリシステム。

【請求項52】 個別にアドレス可能で、かつ、セクタの消去可能ブロックに配置された非揮発性メモリセクタを各々が有する複数の半導体メモリチップであって、前記セクタは各々、メモリ内における自身の物理的位置を定義する物理アドレスを備える該半導体メモリチップと；

メモリへのデータ構造の書き込みおよびデータ構造の読み出しのためのコントローラとを具備するホストプロセッサに接続するためのメモリシステムであって、

コントローラは消去可能ブロックで仮想ブロックを形成し、前記仮想ブロックは、それぞれ、各メモリチップからの消去可能ブロックを含み、コントローラは、仮想ブロックを消去処理されるものと非消去処理されつものちに分別し、かつ、1個の仮想ブロックにデータを充填してから、次にデータを充填させる仮想ブロックへと移動し、仮想ブロックの個々の消去可能ブロックにある1個のメモリセクタに次から次へに書き込みし、それにより異なるチップのセクタに連続して書き込みを行う反復順序でメモリセクタに書き込みすることで各仮想ブロックが充填されるメモリシステム。

【請求項53】 n回の連続セクタ書き込み操作のたびにコントローラがホ

ストプロセッサから受け取った多重セクタ書き込みコマンドを実行し、 n がメモリシステムの半導体メモリチップの数より少ないかまたは等しい場合、コントローラがチップの n それぞれにおける1個のセクタに連続して同時に書き込みする請求項52に記載のメモリシステム。

【請求項54】 個別にアドレス可能で、かつ、セクタの消去可能ブロックに配置され、メモリ内で自身の物理的位置を定義する物理アドレスを持った複数の非揮発性メモリセクタを備える半導体メモリに対して、データ構造の書き込みおよび読み出しを行うコントローラであって、

該コントローラが使用されるメモリシステムのホストプロセッサから受け取った論理アドレスをメモリ内の前記メモリセクタの物理アドレスに翻訳する手段と；

セクタのブロックを消去用と非消去用とに分別する手段と；

ホストプロセッサからのデータが書き込まれるセクタの物理アドレスを指示するライトポインタ（WP）とを備え、

前記ライトポインタ（WP）は、消去用として処理される任意のブロックにおけるメモリセクタの物理アドレスをくまなく所定の順番で移動するように、また、ブロックが満杯の場合は、別の消去ブロックに移動するようにコントローラによって制御され、

ホストプロセッサからセクタ書き込みコマンドを受けると、前記ライトポインタ（WP）がその時点で指示する物理アドレスを論理アドレスに割当てることによって、コントローラはホストプロセッサから受け取った論理アドレスをデータが書き込みされる物理アドレスに翻訳し、

コントローラによって既に割当てられているそれぞれの物理アドレスで論理アドレスのセクタ割当てテーブル（SAT）をコンパイルして、ホストプロセッサからのデータがメモリセクタに書き込みされる頻度より少ない頻度でSATを更新するよう構成されたコントローラ。

【請求項55】 個別にアドレス可能で、かつ、セクタの消去可能ブロックに配置され、しかもメモリ内で自身の物理的位置を規定する物理アドレスを持った複数の非揮発性メモリセクタを有する半導体メモリに対してデータ構造の書き

込みおよび読み出しを制御する方法であって、

セクタのブロックを消去用ブロックと非消去用ブロックとに分別する工程と；

ホストプロセッサからのデータが書き込まれるセクタの物理アドレスを指示する少なくとも1個のライトポインタ（WP）を設け、前記ライトポインタ（WP）が、消去用として処理される任意のブロックにおけるメモリセクタの物理アドレスをくまなく所定の順番で移動するように、また、ブロックが満杯の場合は、別の消去ブロックに移動するように前記少なくとも1個のライトポインタ（WP）を制御し、ホストプロセッサからセクタ書き込みコマンドを受けると、前記ライトポインタ（WP）がその時点で指示する物理アドレスを論理アドレスに割当てることによって、ホストプロセッサから受け取った論理アドレスをデータが書き込みされる物理アドレスに翻訳する工程と；

コントローラによってそれぞれの物理アドレスが既に割当てられている論理アドレスのセクタ割当てテーブル（SAT）を非揮発性半導体メモリに記憶する工程と；

ホストプロセッサからのデータがメモリセクタに書き込みされる頻度より少ない頻度でSATを更新する工程からなる前記の制御方法。

【請求項56】 ホストプロセッサに接続するためのメモリシステムであって、

個別にアドレス可能で、かつ、セクタの消去可能ブロックに配置された非揮発性メモリセクタを有する複数の半導体メモリであって、前記セクタは各々、メモリ内における自身の物理的位置を定義する物理アドレスを備える該半導体メモリと；

データ構造のメモリへの書き込みおよび読み出しのためのコントローラとを具備し、

コントローラはホストプロセッサから受け取った論理アドレスをメモリ内の前記メモリセクタの物理アドレスに翻訳する手段を含み、

前記メモリセクタ（1）は、各々、物理的にデータ領域（300）とスペア領域（302）とに分割され、コントローラは、セクタのデータ領域（300）の

始めからオフセットされるセクタ位置において、ヘッダデータおよびエラー修正コードデータ（ECC）を含むオーバーヘッドデータ（OD）を書き込みし、かつ、ホストプロセッサから受け取ったユーザデータをオーバーヘッドデータ（OD）の何れか一方の側のセクタにおける残りスペースに書き込みするよう構成されているメモリシステム。

【請求項57】 前記オーバーヘッドデータ（OD）は、セクタに書き込みされるユーザデータのうち少なくとも1ビットによって決定される総計によってオフセットされる請求項56に記載のメモリシステム。

【発明の詳細な説明】

【0001】

本発明は、データ記憶および検索のための半導体メモリシステムに関し、また、半導体メモリシステムの不揮発性メモリへのアクセスを制御するメモリコントローラに関する。より具体的には、本発明は、フラッシュメモリシステム及びフラッシュメモリ用のコントローラに関する。

【0002】

フラッシュEEPROM（電氣的消去書き込み可能な読出し専用メモリ）装置は不揮発性データ記憶装置用に電子工業界で一般的に用いられている。NAND型メモリセル、AND型メモリセルあるいはNOR型メモリセルに基づく装置を含む様々なタイプのフラッシュメモリ装置が存在する。こうした装置は、それぞれインタフェースするように設計されたホストプロセッサシステムに対して異なる型のインタフェースを有し、例えば、順次アクセス型インタフェース（多くのNAND型及びAND型装置で一般的に使用）、または、ランダムアクセス型インタフェース（複数のNOR型装置で使用）などが使用されている。本発明は、適切な形で、これらの異なるタイプの記憶装置の少なくともいくつかに、好ましくは全てに利用できることを目的とする。

【0003】

磁気ディスク記憶装置をコンピュータシステムにエミュレーションするために半導体メモリシステムを使用することは周知である。当業界では、半導体メモリシステムの演算速度を増して、磁気ディスク記憶装置をより良くエミュレートすることを目指している。

【0004】

本発明の第1の態様によれば、個別にアドレス可能で、かつ、セクタの消去可能ブロック内に設けられた不揮発メモリセクタを有し、前記セクタは、各々、メモリ内の自身の物理的位置を定義する物理アドレスを含む半導体メモリと；

データ構造のメモリへの書き込みとメモリからの読出し、及び消去用として処理されるブロックと非消去用として処理されるブロックとにセクタのブロックをソートするためのコントローラと；

を具備するホストプロセッサに接続するためのメモリシステムであって、
前記コントローラは、

ホストプロセッサから受け取った論理アドレスをメモリ内の前記メモリセクタの物理アドレスに変換する手段と；

ホストプロセッサからのデータが書き込まれる物理セクタアドレスを指示する書き込みポインタ（以下、ライトポインタ（WP）と称す）であって、消去用として処理される任意のブロックにおけるメモリセクタの物理アドレスを所定の順番で移動するように、また、ブロックが満杯の場合は別の消去ブロックに移動するようにコントローラによって制御されるライトポインタ（WP）とを含み、

こうした構成とすることで、ホストプロセッサからセクタ書き込みコマンドを受け取ると、前記ライトポインタ（WP）が指し示す物理アドレスを論理アドレスに割当てることによって、コントローラはホストプロセッサから受け取った論理アドレスをデータが書き込みされる物理アドレスに変換し、かつ、コントローラによって既に割当てられている各々の物理アドレスとともに論理アドレステーブルをコンパイルし（このテーブルは、以下、セクタ割当てテーブルまたはSATと称す）、コントローラはホストプロセッサからのデータがメモリセクタに書き込みされる頻度より少ない頻度でSATを更新するよう構成されたメモリシステムが提供される。

【0005】

ホストプロセッサからのデータがメモリ内のセクタに書き込まれる度にSATを更新するのではなく、それより少ない頻度でSATを更新することにより、本発明は、例えばフラッシュメモリなどの半導体メモリの超高速処理を提供し、従って、磁気ディスク記憶装置の良好なエミュレーションを可能にする。

【0006】

SATの物理セクタアドレスは、好ましくは、論理セクタアドレスによって命令され、これにより、N番目のSATエントリーは、論理アドレスNを有するデータが書き込みされたセクタの物理アドレスを含む。セクタ読出しコマンドをホストプロセッサから受け取ると、コントローラは、ホストプロセッサから受け取った論理セクタアドレスをSAT内で検索して、過去にコントローラが前記論理

セクタアドレスに割当てた物理セクタアドレスを得る。SATは、好ましくは、半導体メモリのメモリセクタの1または複数の前記ブロック内に記憶される。各ブロックは、以下SATブロックと称されるSATの任意の部分を含んでいる。SATは、SATにおける1つまたは複数のブロックを書き換えることによって更新される。SATセクタのブロック全体を一時に更新することで、メモリシステムの演算スピードが著しく向上する。

【0007】

少なくとも1つのセクタブロック（以下、追加SATブロック（ASB）と称す）が提供され、これは前記SATブロックの個別セクタの修正バージョンを含む。前記ASBブロックの各セクタは、好ましくは、それが更新するSATブロックにおけるセクタの物理アドレスと、前記SATブロックセクタの修正バージョンとを含む。ASBの目的は、SATの半導体メモリ修正セクタに個別に隠して（キャッシュして）、SATブロックの書き換え回数を減らすことにある。前記ASBブロックの全てのセクタがSATセクタの修正バージョンで書き込みされると、ASBブロック内の全ての修正バージョンを含むように個々のSATブロックが書き換えられて、ASBブロックは消去される。

【0008】

本発明のメモリシステムにおいては、ホストプロセッサから受ける任意の論理アドレスに割当てられる物理アドレスが論理アドレスその物に従属していないことは理解されよう。コントローラは、単に、ライトポインタがそのとき指し示す物理セクタアドレスを割当てる。

【0009】

上述したように、コントローラは、ライトポインタ（WP）を他のブロックに移動させる前に、消去処理される前記1つのブロックを充填する。コントローラは、ライトポインタ（WP）を消去処理されるブロック内をくまなく所定の順番で移動させるように好適に構成されている。

【0010】

コントローラは、ライトポインタ（WP）を好適に制御して、消去ブロック中くまなく物理アドレスを番号順に上昇しながら連続して移動させ、このとき、各

ブロックは書き込みされたデータで充填されている。ひとたび物理アドレス順による最高位ブロックのセクタがデータで充填されると、WPはコントローラによって消去処理される全てのブロック中、番号的に最下位の物理アドレスを有するセクタのブロックを取り囲むようにコントローラによって制御されるという意味において、ライトポインタ（WP）の制御は循環式である。

【0011】

コントローラは、あるいは、メモリセクタデータを書き込みするのに別の所定の順番を用いる。例えば、コントローラは、ライトポインタ（WP）を制御して、消去処理されるブロック中を物理アドレスによる番号順に連続して下のように移動させる。もしくは、物理セクタアドレスを非連続順に移動することも可能である。例えば、WPが消去処理される各ブロックにおいて物理セクタアドレス中をアドレスの数字を下るように移動したり、例えば、前記ブロックの各々において、第1セクタの物理アドレスにしたがって番号順に上るなど、何らかの所定の順序でブロックからブロックへ移動したりすることもできる。

【0012】

消去処理されるブロックにおけるセクタにデータを書き込むためにその他にも多くの所定の順番が実行可能なことは理解されよう。さらに、予め決定する必要のない、または、一部のみ決定させた任意の順序でコントローラが消去ブロックを使用することも可能である。一般的には好ましくないが、消去ブロックは、無作為な順序で使用することもできる。

【0013】

前記セクタブロックの個々におけるメモリセクタは、好ましくは、ユニットとして消去可能である。セクタは、また、個別消去も可能である（例えば、半導体記憶装置がAND型メモリの場合）。コントローラは、好ましくは、メモリセクタのブロック全体のみを消去するように記憶装置の消去演算を制御するように構成される。メモリセクタ内の全てのセクタが消去セクタである場合は、セクタのブロックは、消去ブロックとしてコントローラによって処理される。ブロックが1つまたは2つ以上の不良（すなわち、欠陥）セクタを含んでいる場合、コントローラは、ブロック全体を不良と定義し、そのブロックを非消去ブロックとして

処理し、それによりこのブロックへのデータの書き込みは行われない。もしくは、ブロックが1つまたは2つ以上の不良セクタを含んでいる場合、コントローラは、そのブロックを消去ブロックとして処理し、それによりそのブロック内の良好セクタをデータ保存に使用する。しかし、後者の場合、好ましくは、メモリシステムが不良セクタを識別するテーブルを備え、コントローラは、ライトポインタ（WP）が次に移動するセクタアドレスが不良セクタアドレスか否かをチェックし、それが不良セクタのアドレスだった場合、ライトポインタを制御してこの不良セクタをスキップさせ、セクタが書き込みされる所定の順番に従って、次のセクタアドレスに移動するように構成される。

【0014】

疑念を回避するため、既に書き込みされた任意の良好（すなわち、非欠陥）セクタを含む任意のブロックは非消去ブロックとしてコントローラによって処理される。さらに、“消去”セクタなる用語は消去されたセクタのみでなく、書き込みされていない状態のセクタ及び消去前のセクタも含む。従って、書き込み前のセクタのブロックは、コントローラによって消去ブロックとして処理される。

【0015】

好ましくは、セクタブロックは各々、メモリ内での自身の物理的位置を定義する物理ブロックアドレスを有する。前記メモリセクタの各々の物理アドレスは、好ましくは、それが位置するブロックの物理ブロックアドレスを含む。新規なコントローラは、好適に、消去処理されるセクタブロックの少なくとも何個かの物理ブロックアドレスのリストをコンパイルするように構成され、これを利用して次に書き込みが行われるセクタブロックを素早く識別する。この消去ブロックのアドレスリストは、好ましくは、コントローラによってメモリシステムに設けられた一時的記憶域に記憶され、一時的記憶域は、好適に、コントローラのマイクロプロセッサのSRAMであり、各セクタブロックの消去状況を識別するコントローラによって半導体メモリに既に記憶されている情報からもたらされる（この情報は、好ましくは、半導体メモリにおいてビットマップの形式で保持され、各ブロックは消去ブロックまたは非消去ブロックとして記録される）。

【0016】

コントローラを好適に構成することで、以前に別のセクタに書き込みされた旧データを伝えるセクタの書き込みコマンドをホストプロセッサからコントローラによって受けると、コントローラは、現時点の旧データを備えるセクタのアドレスを一時的記憶域に保存する。この一時的記憶域は、好適に、コントローラのマイクロプロセッサに設けられたSRAMまたはDRAMである。通常はユーザによって発せられる、セクタ削除コマンドをホストプロセッサからコントローラによって受け取ると、コントローラは、好ましくは、削除すべきセクタを廃棄としてマークする（このとき、物理的にセクタを消去することはない）。コントローラは、ライトポインタ（WP）によって書き込みされた旧データを含む1つまたは2つ以上のセクタを有するブロック（以下、カレントオブソリートブロック（COB）と称す）をいかなる場合も1個しか容認しない。COB内の全てのセクタが旧データを含むと、COBは即座に消去される。これは、ライトポインタ（WP）が次のブロックに移動する前に、消去処理される各ブロック内のメモリセクタアドレスを連続して移動する場合に特に好都合である。このような場合、削除される一連の廃棄セクタ（例えば、書き換えられたユーザデータファイルの一部を含む）は、多くの場合、全て同一のブロック内にある。一連のセクタが、それらが過去に書き込みされた順序と違う順番で書き換えられると、2個以上のブロックに廃棄セクタが生じることがある。COB以外のブロックにあるセクタが旧データを含む場合、コントローラは、好ましくは、COB内の有効（非廃棄）セクタにある任意のデータを、ライトポインタ（WP）がその時点で指し示す別のブロックに再配置し、その後COBを消去する。COB以外のブロックにある前記セクタが、その後、廃棄としてマークされ、このブロックが今度はCOBになる。ライトポインタの現在位置に再配置したデータを書き込みするというよりむしろ、メモリシステムは、そうしたリロケーションデータを書き込むセクタの物理アドレスを指示するための第2のライトポインタ（以下、リロケーションポインタ（RP）と称す）を備え、リロケーションポインタ（RP）は、常にライトポインタ（WP）とは異なるセクタブロックにある。これは、ホストプロセッサによって書き込みするように直接命令された、すなわち、ライトポインタ（WP）によって書き込みされるデータ構造と、リロケーションデータとが交じり合

うことを阻止するという効果をもたらす。

【0017】

通常、2つのタイプのデータのみがホストプロセッサから半導体メモリに書き込みされる。すなわち、ファイルデータとシステムデータである。再配置及び消去の回数をさらに減じるために、本発明のメモリシステムは、第3のライトポインタ（以下、システムライトポインタ（SWP）と称す）をさらに備える。これはホストプロセッサからのシステムデータを書き込みするセクタの物理アドレスを指示し、SWPは、常に、ライトポインタ（WP）とは別のブロックにある（さらに、リロケーションポインタがある場合は、これとも違うブロックにある）。好ましくは、システムデータは、システムの初期化中に識別され、かつ、必要に応じて演算中に更新される。

【0018】

ライトポインタ（WP）とシステムライトポインタ（SWP）の両方が備えられている場合、ファイルデータは、常にライトポインタ（WP）によって指示されたアドレスに書き込みされる。好ましくは、リロケーションポインタ（RP）とシステムライトポインタ（SWP）の双方がライトポインタ（WP）に類似の方法で消去処理される前記ブロックにおけるメモリセクタの物理アドレス中を移動するように制御される。従って、前記ブロックの（良好）セクタの全てがリロケーションデータまたはシステムデータで充填された場合、対応するリロケーションポインタ（RP）およびシステムライトポインタ（SWP）の一方が、消去処理されるブロックにおける全セクタの物理アドレスからコントローラによって使用されるように定義された次のアドレスに移動する。

【0019】

システムライトポインタ（SWP）が設けられている場合、好ましくは、コントローラは、1つまたは2つ以上の廃棄セクタを含む少なくとも2個のブロックが任意のときに存在することを容認する。このとき、一方は前記COBであり、他方は、1つまたは2つ以上の旧システムデータセクタを含むカレントオブソリートシステムブロック（COSB）である。COSBの消去を容認するために任意のシステムデータセクタを再配置することが必要になった場合、好ましくは、

再配置されたシステムデータはシステムライトポインタ（SWP）がその時点で指示するアドレスに送られる。

【0020】

事実、任意の一時に旧データを有する2個以上のブロック（COB及びCOS B）が一時的に存在することがある。例えば、COBの消去が必要な場合（旧データがちょうど別のブロックに作り出された後）、ライトポインタ（WP）の1つはこれを指し示す。すなわち、WPは、現在COBであるブロックにまだ書き込みを行っている。このような場合、コントローラは、好ましくは、新しいCOBの作成処理を行い、一方、コントローラの定義に従って、古いCOB内にある全ての消去セクタが充填され、かつ、ライトポインタ（WP）が次に使用する消去ブロックに移動するまで古いCOB（以下、保留オブソリートブロック（POB）として扱う）の消去を保留する。このとき、POB内の任意の有効（非廃棄）データは再配置され、POBが消去される。

【0021】

ホストプロセッサからのデータ構造をメモリに書き込みすることに加えて、コントローラは、制御情報として指定されたメモリデータの生成および書き込みを行う。好ましくは、コントローラは、そうした制御情報をホストプロセッサから受け取ったデータ構造が書き込まれたメモリのセクタブロックとは別個のブロックに書き込む。こうした制御情報を記憶するためのブロックを以下、制御ブロック（CBs）と称する。これらの制御ブロックはコントローラによって周期的に更新され、初期化中、および時にメモリシステムの演算中にアクセスされる。

【0022】

好ましくは、SATが最後に更新された後、コントローラは、ライトポインタ（WP）によって書き込まれたデータ構造用の論理セクタアドレスのリストを一時的記憶域（これは、メモリシステムに設けられたRAM、または、コントローラのマイクロプロセッサに好適に埋め込まれたSRAMあるいはDRAM）に記憶させる。SRAMに記憶されたこのリストは、以下、WSLの論理アドレスとして言及される。これは、メモリ内の不揮発性セクタに書き込みされた順番に好適に記憶される。連続して書き込みされたセクタ群にとって好都合なことに、W

S L エントリーは、従って、第1セクタの論理アドレスとして、かつ、セクタ群の長さ、すなわち、セクタが書き込みされた数だけ書き込みされる。前記セクタ群の各々は、セクタの1つのブロックより広がらないように定義される。

【0023】

コントローラは、また、S A T の最終更新の後、好適に、ブロックがデータ書き込み用にライトポインタ (W P) によって使用された順番を前記一時的記憶域に記憶する。これは、アドレスがW S L に保存された更新されたセクタが位置するブロックのブロックアドレス表の形式で保存される。このブロックアドレス表は、以下、書き込みブロックリスト (W B L) として言及する。メモリシステムは、W S L およびW B L によって、前記連続して書き込みされたセクタ群の第1論理アドレスに割当てられた物理メモリ内に記憶場所の知識を有するものであるから、コントローラは、W S L 及びW B L を使用して、S A T の最終更新の後、前記連続して書き込みされたセクタ群における各論理セクタアドレスの各々について、常に正しい物理セクタにアクセス可能である。好ましくは、W S L は所定の大きさを有し、ひとたびW S L が一杯になると、1つまたは2つ以上のS A T ブロック (及び／又はA S B s) が更新され、W S L とW B L とが空になる。

【0024】

好ましくは、物理セクタアドレスの開始と、最後のS A T 更新後にコントローラによってデータが書き込みされたセクタを有するブロック間のリンクとが、半導体メモリの制御ブロックに記憶される。各セクタに記憶されたユーザデータ用の論理セクタアドレスを、セクタそれ自身、例えばセクタに設けられたヘッダフィールドに記憶することにより、システムに対する任意の電力の除去および回復後、W S L およびW B L を容易に再構成できる。これは、一杯になっていないブロックに到達するまで、半導体メモリをくまなく走査し、S A T が最後に更新されてから書き込まれたセクタ内の論理アドレスを読出すことによって達成される。これは、電力の除去あるいは損失前にライトポインタ (W P) を含むブロックである。これにより、メモリシステムから予期せぬ電力除去が生じた際にも、高度なデータセキュリティが提供される。

【0025】

リロケーションポインタおよびシステムライトポインタがメモリシステム内に含まれている場合、コントローラは、移動されたデータもしくはシステムデータが書き込まれたメモリ内のセクタに対応する類似の論理セクタアドレスリストを、前記一時的記憶域に記憶することが望ましい。以下、これらのリストをそれぞれ、再配置セクタリスト（RSL）、ライトシステムセクタリスト（WSSL）と称する。また、コントローラは、RPおよびSWPによって使用され且つライトブロックリストに類似する対応するブロック順位リストを前記一時的記憶域内に記憶しても良い。以下、これら2つのリストを、再配置ブロックリスト（RBL）およびライトシステムブロックリスト（WSBL）と称する。また、物理セクタアドレスの開始と、最後のSAT更新後に移動データもしくはシステムデータが書き込まれたセクタを有するブロック間のリンクとが、半導体メモリの前記制御ブロック（CBs）の少なくとも1つに記憶されても良い。これにより、メモリを単に走査し且つ最後のSAT更新後にRPおよびSWPのそれぞれによって書き込まれたセクタ内の論理アドレスを読み込むことによって、ホストプロセッサに対する電力の除去および回復後に、RSLおよびWSSLを再構成することができる。

【0026】

前述した任意の実施例における前記各セクタは、メモリの単一の「ページ」、すなわち、メモリセクタにおける前記ブロック内の1列のメモリセルから成る。しかしながら、本発明は、そのようなセクタフォーマットのみに限定されるものではなく、時として（例えば、NOR型のRAMを使用する場合）、前記各セクタが1ページよりも少なく、あるいは、1ページよりも多くても良い。また、後者の場合、前記セクタの全てが必ずしも同じサイズである必要はない。例えば、本出願人の過去の国際特許出願PCT/GB99/00188号に開示されているようなデータ編成スキームをコントローラによって使用して、半導体メモリ中に存在する個々の欠陥を避けるように適切なサイズ（サブセクタサイズ）のセクタを形成することができる。

【0027】

前述したように、各セクタは個々にアドレス可能である。各セクタは、それぞ

れ個々にアクセス可能な複数のセクタ部を備え、コントローラが、各セクタ部に対して個々に書き込んだり、各セクタ部から個々に読み込んだりしても良い。無論、可能な最も小さいセクタ部のサイズが、アドレス可能な最小のメモリユニットである。NOR型メモリにおいて、例えば、アドレス可能な最小のメモリユニットは、一般に、1バイトである。

【0028】

好ましくは、コントローラは、均一にサイズ設定されたデータセグメント内のメモリセクタに対してデータを書き込み、また、前記メモリセクタからデータを読み込む。全てのメモリセクタが同じサイズである場合、前記各データセグメントは、前記メモリセクタのサイズと同じサイズであることが望ましい。各データセグメントは、ホストプロセッサからのデータ構造（例えば、ファイルまたはシステムデータ）や、コントローラによって形成されるデータを備えていても良い。

【0029】

半導体メモリがNAND型デバイスに基づく場合、コントローラは、不良セクタを含む不揮発性メモリ内のブロックのブロックアドレスリスト（以下、不良ブロックリスト（BBL）と称する）を、1または複数の前記制御ブロックに記憶することが望ましい。この場合、コントローラは、そのようなブロックのそれぞれを「非消去」ブロックとして処理する。これにより、そのようなブロックは、一時的記憶域内に記憶される消去ブロックのリスト内に現れず、コントローラは、データをそのブロックに書き込まない。

【0030】

メモリがAND型デバイスに基づく場合、コントローラは、任意の不良セクタのアドレスリストを、1または複数の前記制御ブロック（CBs）に記憶することが望ましい。この場合、コントローラは、少なくとも1つの不良セクタを含む任意のブロック内の良好なセクタを使用するため、また、任意の不良セクタをスキップするために、1または複数の前記ライトポインタ（WP）を制御する。無論、1または複数の不良セクタを含む1つのブロックが消去される後者の場合、そのブロック内の良好（すなわち、非欠陥）なセクタは、ブロック消去操作中に

個々に消去される。

【0031】

また、コントローラは、好適に、全てのSATブロックのブロックアドレスのリストを、1または複数の前記制御ブロック内に記憶する。このリストは複数のリスト部分から成ることが望ましく、以下、各リスト部分をテーブルブロックリスト(TBL)と称する。前記各リスト部分は、論理的に連続するSATブロックおよび対応する任意のASBsから成る群のブロックアドレスを含んでいる。

【0032】

コントローラは、1または複数の前記制御ブロックを、以下にブートブロック(BB)と称する専用のメモリブロック内に記憶することが望ましい。また、データセキュリティのために必要な他の重要な情報は、ブートブロック内、例えば不良ブロック(あるいは、不良セクタ)のリスト内に記憶されても良い。好ましくは、不良セクタを全く含まないメモリセクタの第1のブロックが、ブートブロック(BB)として指定される。

【0033】

好ましくは、コントローラは、全ての良好なセクタを含むブロックを、単に、SATブロック、制御ブロック、ASBs、BBsとして使用する。

【0034】

一時的記憶域(例えば、コントローラマイクロプロセッサ内のSRAMやDRAMのような、メモリシステム内のRAM)内にキャッシュが設けられても良い。この場合、コントローラは、SATから(コントローラによって)最も新しくアクセスしたSATエントリーを含む連続するSATエントリーの群をキャッシュ内に記憶する。また、これによって、アドレス変換速度が向上する。また、アドレス変換速度の向上は、全てのASB及びこれらが関連付けられるSATブロックの物理アドレスのリストであって、SATセクタ書き込み操作が実行される度に更新されるリスト(以下、ASBリストすなわちASBLと称する)を前記一時的記憶域内に形成することによって達成されても良い。同様に、非常に高速の論理-物理セクタアドレス変換を可能にするため、制御ブロック内のTBLの部分は、前記一時的記憶域に記憶されても良い。

【0035】

半導体メモリは、単一のメモリチップの形態を成す単一のメモリアレーを備えていても良く、あるいは、複数のメモリチップの形態を成す複数のメモリアレーを備えていても良い。メモリが複数のチップから成る場合、コントローラは、好適に、複数のメモリチップ内のメモリセクタを、多数の仮想ブロックへと形作る。前記各仮想ブロックは、前記各メモリチップから消去可能な1つのメモリセクタブロックを備え、好ましくは、コントローラは、前記仮想ブロックを、消去処理されるブロックおよび消去処理されないブロックにソートする。コントローラは、好ましくは、消去処理される仮想ブロックのリストを備えるとともに、これを、コントローラのマイクロプロセッサのSRAMであっても良いメモリシステム内の一時的記憶域に記憶する。コントローラは、連続セクタ書き込み操作のそれぞれにおいて、ライトポインタ(WP)(設けられている場合には、RPおよびSWP)を1つのチップから他のチップへと移動するように制御することが望ましい。この場合、ライトポインタは、仮想ブロックの消去可能な1つのブロック内の1つのセクタから移動し始め、1つのセクタが仮想ブロックの消去可能な各ブロック内に書き込まれるまで、仮想ブロックの消去可能な他の各ブロック内の1つのセクタへと連続的に移動するとともに、その後、最初のセクタが書き込まれたチップへと戻り、同様の方法で進んで、仮想ブロックの消去可能な各ブロック内の他の1つのセクタを一杯にする。その後も同様に移動し、結果的に、仮想ブロックをデータで一杯にする。その後、ライトポインタ(WP)は、消去処理される前記仮想ブロックリスト内の次の仮想ブロックへと移動して、この次の仮想ブロックを同様の方法によって一杯にする。コントローラは、それ自身が実行するn回の連続セクタ書き込み操作(この場合、nは、メモリシステム内の半導体メモリチップの数以下)の度に、n個の各チップ内の1つのセクタに対して略同時に書き込むことができるように構成されていることが望ましい。好ましくは、コントローラは、仮想ブロック内の消去可能な全てのブロックを同時に消去することによって、任意の仮想ブロックの消去を実行する。

【0036】

言うまでもなく、メモリシステムのコントローラは、コントローラデバイスと

して回路内に実質的に実装されていても良いが、少なくとも部分的には、コントローラデバイスのメモリ内に保持されるファームウェアとして具現化されることが望ましい。コントローラは、半導体メモリとして、同じチップ（あるいは、同じ複数のチップのうちの1つ）上に一体形成されていても良い。

【0037】

本発明の第2の態様によれば、複数の半導体メモリチップを備え、各チップは、個々にアドレス可能で且つセクタの消去可能なブロック内に配置される複数の不揮発性メモリセクタを有し、前記各セクタは、メモリ内におけるその物理的な位置を規定する物理アドレスを有する半導体メモリと；

データ構造をメモリに書き込み、また、データ構造をメモリから読み込むためのコントローラと；

を具備するホストプロセッサに接続するためのメモリシステムであって、

前記コントローラは消去可能なブロックを仮想ブロックへと形作り、前記各仮想ブロックは各メモリチップから消去可能なブロックを備え、コントローラは、仮想ブロックを、消去処理されるブロックと消去処理されないブロックとにソートし、コントローラは、満たすべき次の仮想ブロックへと移動する前に、1つの仮想ブロックをデータで満たし、各仮想ブロックは、仮想ブロックの消去可能な各ブロック内の1つのメモリセクタにコントローラが次々と書き込む繰り返しシーケンス（循環シーケンス）により、そのメモリセクタに書き込むことによって満たされ、これにより、異なるチップにセクタが連続的に書き込まれるよう構成されたメモリシステムが提供される。

【0038】

好ましくは、コントローラは、ホストプロセッサから得られる複数のセクタ書き込みコマンドによって実行するn回の連続セクタ書き込み操作（この場合、nは、メモリシステム内の半導体メモリチップの数以下）の度に、n個の各チップ内の1つのセクタに対して略同時に書き込むことができるように構成されている。

【0039】

本発明の第3の態様によれば、個々にアドレス可能で且つセクタの消去可能な

ブロック内に配置される不揮発性メモリセクタを有する半導体メモリに対してデータ構造を書き込むとともに、前記半導体メモリからデータ構造を読み込み、メモリ内におけるその物理的位置を規定する物理アドレスを前記各セクタが有するコントローラが提供される。このコントローラは、

コントローラが使用されるメモリシステムのホストプロセッサから受け取った論理アドレスを、メモリ内の前記メモリセクタの物理アドレスに変換するとともに、セクタのブロックを、消去処理されるブロックと消去処理されないブロックとにソートするための手段と；

書き込まれるセクタの物理アドレスをホストプロセッサから指し示すライトポイント（WP）とを有し、前記ライトポイント（WP）は、コントローラによって制御されて、消去処理される任意のブロック内のメモリセクタの物理アドレスを介して所定の順序で移動し、ブロックが一杯になると、他の消去ブロックへと移動し、

セクタ書き込みコマンドがホストプロセッサからコントローラによって受けられると、コントローラは、前記ライトポイント（WP）が指し示す物理アドレスを論理アドレスに割当てることによって、ホストプロセッサから受け取った論理アドレスを、データが書き込まれる物理アドレスに変換し、また、そのためにコントローラによって割当てられている各物理アドレスとともに論理アドレスのテーブルをコンパイルするとともに、ホストプロセッサからのデータがメモリセクタに書き込みされる頻度より少ない頻度でSATを更新するようにコントローラが構成されている。

【0040】

本発明の第4の態様によれば、個々にアドレス可能で且つセクタの消去可能なブロック内に配置される不揮発性メモリセクタを有する半導体メモリに対してデータ構造を書き込むとともに、前記半導体メモリからデータ構造を読み込み、メモリ内におけるその物理的位置を規定する物理アドレスを前記各セクタが有する方法が提供される。この方法は、

セクタのブロックを、消去処理されるブロックと消去処理されないブロックとにソートし；

書き込まれるセクタの物理アドレスをホストプロセッサから指し示すライトポインタ (WP) を提供し;

少なくとも1つの前記ライトポインタ (WP) を制御して、消去処理される任意のブロック内のメモリセクタの物理アドレスを介して所定の順序でライトポインタを移動するとともに、ブロックが一杯になると、他の消去ブロックへとライトポインタを移動させ;

セクタ書き込みコマンドがホストプロセッサからコントローラによって受けられると、前記ライトポインタ (WP) が指し示す物理アドレスを論理アドレスに割当てることによって、ホストプロセッサから受け取った論理アドレスを、データが書き込まれる物理アドレスに変換し;

変換のためにコントローラによって割当てられている各物理アドレスとともに、論理アドレスのテーブルを、不揮発性半導体メモリ内に記憶し;

ホストプロセッサからのデータがメモリセクタに書き込みされる頻度より少ない頻度でSATを更新する。

【0041】

以下、添付図面を参照しながら、本発明の好ましい実施例を単なる一例として説明する。

【0042】

図1は、NAND型メモリセルを基本とするフラッシュメモリアレーの1ブロック4における物理的なページ構造を概略的に示している。図1は、ブロック4内の3つのページ1, 2, 3を示している。物理的な観点からすれば、ページ1は、メモリブロック内の1列のメモリセルから成る。メモリは、複数のブロックに分割されており、各ブロックは、メモリセルの多数の列 (すなわち、多数のページ) から成る。各ページ1, 2, 3は、後述する528バイトのフラッシュメモリシステムにおいて、物理的なメモリ空間の1セクタとして扱われる。メモリ内の各ページ1は、個々にアドレス可能 (読み込み/書き込み、および、オペレーションの削除) であり、また、これらのページはブロック内で消去することができる。以下、そのようなメモリアレーを組み込むメモリシステムについて説明する。また、AND型もしくはNOR型のフラッシュメモリに基づくメモリシステ

ムについても、別途、後述することにする。

【0043】

図2は、フラッシュメモリチップ5およびコントローラチップ8を組み込んだメモリシステム10を示している。フラッシュメモリチップ5は、フラッシュメモリアレー6と、コントローラチップ8のコントローラバッファ9に接続されたリード／ライトバッファ7とを備えている。また、コントローラチップ8は、コントローラマイクロプロセッサ11と、エラー修正コード（ECC）ジェネレータチェッカ12とを有している。コントローラバッファ9はホストコンピュータプロセッサ（図示せず）に接続しており、ホストコンピュータプロセッサは、コントローラチップ8の出力部O／Pを介して、メモリシステム10に接続されている。コントローラチップ8（以下、コントローラと称する）は、メモリアレー6に対するデータ構造の読み込み及び書き込みを制御する。メモリシステム10に接続されたホストプロセッサ2は、読み込み及び書き込みコマンドを、コントローラ8に送る。データは、512バイト部分すなわち「ホストデータセクタ」の状態で、ホストによりアクセス可能であり、各ホストデータセクタは論理セクタアドレス（LA）を有している。コントローラ8は、ホストプロセッサからLAを受けて、これを後述するように物理アドレスに変換する。この場合（NAND型メモリの場合）、各物理アドレス（PA）は、アレー6内のフラッシュメモリのページ1の物理的な位置を規定する。各LAは、1つの24ビット領域の形態を成している。LAを使用してPAを入手することは、アドレス変換と称されており、一般に、リード／ライドのアクセスを行なう度に必要となる最も頻繁な演算である。コントローラ8は、データセグメントの状態でデータをメモリアレー6に書き込む。各セグメントは528バイトの大きさを有している。ホスト（例えば、ユーザファイルまたはシステムデータ）から受け取った512バイトの各データのため、コントローラは、マイクロプロセッサ11によって形成された4バイトのヘッダーと、ECCジェネレータチェッカ12によって形成された12バイトのECCとから成る16バイトのデータを形成する。コントローラは、フラッシュバッファ7を介して、この16バイトのデータを、メモリアレー6の1ページに書き込まれる528バイトのデータセグメントへと編成する。

【0044】

ホストデータセクタの論理アドレス (LA) は、4 バイトのヘッダーの状態で、ホストデータセクタが書き込まれるフラッシュセクタ 1 内に記憶される。読み込み操作時、フラッシュメモリアレーの関連するセクタ内に記憶されたデータは、フラッシュリード/ライトバッファ 7 を介して、アレー 6 からコントローラバッファ 9 へと読み込まれる (同時に、データ内のエラーをチェックするために、EC C ジェネレータチェッカへと読み込まれる)。そして、コントローラは、記憶された LA がホストコンピュータによって要求されている LA と一致していることをチェックするため、ホストコンピュータによるコントローラバッファ 9 からのデータの読み込みを許容する前に、4 バイトのヘッダーを読み込む。

【0045】

コントローラ 8 は、個々のセクタベース上でメモリ 6 に書き込まれるデータの物理的な場所を管理する。更に詳細に後述するように、コントローラは、消去されたブロックのビットマップをメモリ 6 内に記憶するとともに、消去されたブロックが書き込みのために使用されるブロックの物理アドレスの昇順で順序付けられた少なくとも幾つかの消去されたブロックのリスト (ネクストイレーズドブロック (Next Erased Block (NEB)) リスト) を、マイクロプロセッサ 11 内の SRAM 内にコンパイルする。ホストデータセクタが書き込まれる物理的なページの場所は、ホストから受ける論理アドレスに依存しない。各ホストデータセクタは、サイクリックライトポイントによって規定されるアドレスに書き込まれる。異なるタイプの書き込み操作のため、特定のライトポイントが使用される。すなわち、ホストファイルデータの書き込みは、データライトポイント (WP) によって指し示されたアドレスで実行され、ホストシステムデータの書き込みは、システムライトポイント (SWP) によって指し示されたアドレスで実行される。リロケーションポイント (RP) は、ホストによって直接に順序付けられなかったセクタを書き込むために使用される。これらの各ライトポイントは同じ性質を有している。すなわち、各ポイントは、ブロックの複数のページにわたって連続的に移動し、その後、ネクストイレーズドブロック (NEB) リスト内の次に消去されたブロックの第 1 のページへと移動する。消去されないファイルを含むブロックは、「

非消去」ブロックとして処理され、ポインタが1つのブロックから他のブロックへと移動する際に読み飛ばされる（そして、NEB内に含まれることはない）。

【0046】

セクタリロケーションアルゴリズム

セクタのブロック4を消去して、旧データを含むセクタ空間を回復させる場合には、ブロックを消去できるように、有効セクタと廃棄セクタとの組み合わせを含むブロックからセクタを再配置しなければならない。原理的には、コントローラ8により、特定のライトポイントに対応する1つのブロックだけが、旧データセクタを常時収容することができる。ホストによって書き込まれるセクタが第2のブロック内で廃棄セクタを形成する場合には、有効セクタを再配置した後、必要に応じて、既存のブロックをまず最初に消去しなければならない。

【0047】

したがって、大部分のブロックが有効セクタと廃棄セクタとを有している場合には、多数のセクタの消去および広範囲にわたるセクタの再配置は避けられない。これは、ファイルの一部としてホストにより書き込まれたセクタの順序が、以前にセクタが書き込まれた順序と異なる場合にだけ、生じる。これは、殆どのアプリケーションにおいて、通常のケースではない。しかしながら、通常のファイルの書き込み操作の場合であっても、関連しないデータを、ファイルの「先端（head）」や「末端（tail）」を含むブロックからリロケーションしなければならない。他のファイルデータと混合されたシステムデータの再配置によって他のブロックの消去が別途に引き起こされ、これによって、このブロックからの更なる再配置が引き起こされるといった可能性が高い。

【0048】

したがって、再配置および消去の総数を減少するために、システムデータは、厳密に識別され、常に、システムライトポイント（SWP）のアドレスに書き込まれ或いはリロケーションされる。システムデータに関する情報は、初期化プロセス中に得られ、マイクロプロセッサのSRAM13内に記憶される。無論、データファイルは、ホストコンピュータプロセッサ内のファイルシステムによって、フラッシュメモリに書き込まれる。ファイルデータは、ファイルシステムによって、

複数のクラスタに分割される。この場合、各クラスタは、（一般には）512バイトの連続ホストデータセクタの群である。ファイルシステムは、メモリに記憶されたファイルの属性および各ファイルを形成するクラスタの場所に関するテーブルおよびデータ構造を維持する。これらのテーブルおよび構造は、フラッシュメモリ内に（システムデータとして）記憶されるとともに、ファイルデータが書き込まれる場合にはいつでも、ファイルシステムによって更新される。ファイルデータは、メモリに記憶される際、ファイルに関連するシステムデータ（ディレクトリおよびFAT（file allocation table）に関する）を伴う。一般に、メモリに書き込まれるシステムデータは、BIOSパラメータコンフィギュレーション情報、各エントリが特定のクラスタに関連するファイルアロケーションテーブル（FAT）の1つ又は2つのコピー、ルートディレクトリ、サブディレクトリを含んでいる。コントローラは、ホストシステムデータセクタを書き込むための操作を認識するように構成されており、これにより、このホストデータセクタを、ファイルデータのホストデータセクタと異なるように取り扱うことができる。システムセクタの書き込みを認識するため、以下のように、多数の方法を単独で或いは組み合わせて使用しても良い。

1. システムデータは、単一のセクタ書き込みコマンドを用いて書き込まれる。一方、ファイルデータは、複数のセクタ書き込みコマンドを用いて書き込まれるても良い。

2. ファイルシステムルートディレクトリ内で最後のセクタアドレスよりも下位にあるLAを有する全てのセクタが、システムセクタである。このアドレスは、ホストファイルシステムによってメモリ内に記憶されたBIOSパラメータブロック内に保持された情報から決定することができる。

3. サブディレクトリ内の全てのセクタがシステムセクタである。サブディレクトリアドレスおよびサイズは、全てのルートディレクトリおよびサブディレクトリのエントリを読み込むことによって認識することができる。

4. システムセクタは、しばしば、それらが再度書き込まれる直前に、ファイルシステムによって読み込まれる。

【0049】

同じ目的のため、再配置されるファイルデータセクタは、リロケーションポイント（RP）によって規定されるアドレスに書き込まれ、したがって、ホストによって書き込まれたセクタと混合されない。

【0050】

本発明の変形例においては、再配置されたシステムデータが書き込まれる場所を指し示すために、別のライトポイントを準備しても良い。この別個のポイントは、システムリロケーションポイント（SRP）と称され、常に、WPおよびSPと異なるブロック内に配置される

【0051】

ブロック消去アルゴリズム

本発明において、消去されるブロックの任意選択や、バックグラウンド消去のスケジューリングは、全く実行されない。第2のブロック内の廃棄セクタが、保留のホストセクタ書き込みコマンドによって生じる場合には、通常、廃棄セクタを含むブロックのリスト消去が即座に実行される。同様に、制御ブロックの書き換えの結果として、ブロックが廃棄制御データを全体として含む場合には、ブロックは即座に消去される（詳しく後述するが、制御ブロックは、コントローラ8が特定の制御データを書き込む場所である）。

【0052】

したがって、通常、再配置によって旧データを形成することはできないため、旧データを有する存在可能なブロックは、たった2つにすぎない。すなわち、データライトポイント（WP）に対応し且つ廃棄ファイルデータを含むカレントオブソリートブロック（Current Obsolete Block(COB)）、および、システムライトポイント（SWP）に対応し且つ廃棄システムデータを含むカレントオブソリートシステムブロック（Current Obsolete System Block(COSB)）だけである。しかしながら、各タイプのもう1つの廃棄ブロックが一時的に存在する場合がある。これは、消去されるブロック（旧データが他のブロック内に形成された）がその時点で任意のタイプのライトポイントを含んでいる場合に生じる。この場合、そのようなブロック（保留オブソリートブロック（Pending Obsolete Block (POB)）と呼ばれる）の消去は、このブロック内の全ての消去ページが使用され

且つ関連するライトポイントが他のブロックへ移動されるまで、延期されなければならない。この時点で、保留の廃棄ブロックが即座に消去される。

【0053】

前述したように、消去されたブロックの同一性は、全てのフラッシュブロックアドレス空間を測るビットマップ (MAP) 内で維持され、このMAP内に、各ブロックの消去状態が記録される。セクタもしくは制御データをブロックアドレスの順序で連続的に書き込むために、消去されたブロックは破壊される。バックグラウンド消去は実行されない。1 または複数の不良セクタを含む任意のブロックは、不良ブロックとして処理されるとともに、コントローラによって「非消去」ブロックとして処理される。

【0054】

ウェアレベリング (Wear Levelling)

サイクリックライトポイントを使用して単一のセクタ書き込み管理を行なうと、フラッシュメモリ内に固有のウェアレベリングが生じる。しかしながら、ブロックを消去するアルゴリズムは、旧データもしくは消去済みデータとともに存在すると直ちに、セクタ書き込み操作のシーケンスの機能であるウェアレベリング特性を形成する。任意の更なるウェアレベリングが必要であると思われる場合には、ランダムブロック内でセクタを一時的に再配置してこれらのブロックを消去できるようにする別個の更なる技術を組み込んでも良い。

【0055】

アドレス変換原理

主なアドレス変換手段は、基本的に、論理アドレスによって順序付けられたセクタの物理アドレスのリストであるセクタアドレステーブル (SAT) である。したがって、N番目のSATエントリは、通常、論理アドレスNを有するセクタのための物理アドレスを含んでいる。SATは、多数の独立ブロック (SATブロック) として編成されるとともに、SATブロックの個々のページを書き換えることによって更新される。SATブロックは、これとリンクされてSATブロックの個々のページの修正ができる専用の追加SATブロック (ASB) を有していても良い。セクタ書き込み性能に対する影響を最小限にするために、SAT

ページは、各セクタ書き込み後に書き換えられない。無論、頻繁には書き換えられない。

【0056】

したがって、SATは、最後に更新されてから、書き込まれたセクタに関する正確な物理アドレスを含んでいない。そのようなセクタの論理アドレスは、ライトセクタリスト (WSL)、再配置セクタリスト (RSL)、ライトシステムセクタリスト (WSSL) と呼ばれるリストの状態で、プロセッサによって、そのSRAM13内に記憶される。これらのリストは、ホストによって書き込まれた或は消去前にブロックから再配置されたセクタの順序と正確に一致している。連続的にセクタが書き込まれる場合、WSLおよびRSLエントリーは、第1のセクタ論理アドレスおよびセクタ群長さを規定する。セクタ群は、1つのブロックから他のブロックへとジャンプすることができない。マイクロプロセッサ11は、フラッシュメモリ内のセクタ系列の始点と、セクタ書き込みのためのブロックが使用される順番とに関する情報 (前述したリストに対して相補的で且つプロセッサのSRAM内に形成される特定のリスト、すなわち、ライトブロックリスト (WBL)、ライトシステムブロックリスト (WSBL)、再配置ブロックリスト (RBL) は、後述するが、この情報を記憶するために使用される) を有している。

【0057】

WSL、RSLおよびWSSL (そして、これらに相補的なリスト、すなわち、WBL、WSBL、RBL) は、最後のSATが書き換えられた後に書き込まれた系列内のセクタのヘッダー内で論理アドレスを読み込むことによってメモリシステム10に対して電力を除去および回復した後、マイクロプロセッサ11によって再構成され得る。系列および系列内のセクタを含むブロック間のリンクにおける開始セクタアドレスは、マイクロプロセッサにより、フラッシュメモリ内の制御ブロック (CB) と呼ばれる特定のデータ構造内のエントリーから得られる (制御ブロック (CB) については後述する)。この方法によれば、カードから予期しない電力の除去が生じた際には、データの高い安全性を確保できる。

【0058】

無論、前述したように、システムリロケーションポイント（SRP）が含まれている場合には、システムリロケーションセクタリスト（SRSL）および相補的なシステムリロケーションブロックリスト（SRBL）が形成され、これらは、WSL、RSL、WSSL、および、WBL、RBL、WSBLのそれぞれに関し、前述したと同様の方法で使用される。

【0059】

コントローラ8によってデータが書き込まれるフラッシュメモリセクタ（すなわち、ページ）1内の本実施例に係るデータ構成が図3に示されている。セクタ1は、まず、例えばホストデータセクタから成る512バイトの情報部分1aを含んでおり、それに続いて、4バイトのヘッダー部分1bおよび12バイトのECC1cを順に含んでいる。図4に示されるように、ヘッダー部分は、それ自身、データ構造タイプ部分20と、ヘッダーパラメータ22（例えば、情報部分1a内に書き込まれたホストデータセクタの論理セクタアドレス（LA）から成る）とを備えている。データ構造タイプは、データセクタ、削除データセクタ、セクタアドレステーブル（SAT）ページ、追加SATブロック（ASB）ページ、制御ブロック（CB）ページ、ブートブロック（BB）ページのいずれか1つを示す値を有することができる。

【0060】

削除データセクタ

削除データセクタは、物理的には、フラッシュメモリ内で、廃棄もしくは削除セクタデータすなわちCOBもしくはCOSBを含むことが許容されるブロックにだけ一時的に存在する。それは、ヘッダー内のデータ構造タイプ領域の「オールゼロ」状態、あるいは、適当な場合には他の手段、によって認識される。

【0061】

セクタアドレステーブル（SAT）

SATは、論理セクタの物理アドレスを含む一連のエントリーである。エントリーNは、論理セクタNのための物理アドレスを含んでいる。ページ内のエントリーは、ページ内に記憶されたデータの512バイトの情報部分1aを占めている。1つの操作でSATページが書き込まれるため、全ページを保護するために

E E C領域を使用でき、S A Tページ内の各エントリーのための別個のE C C領域が不要となる。

【0062】

実際に、S A Tは一連のブロックとして記憶され、各ブロックは最大で64のS A Tページを有している。別個のデータ構造およびテーブルブロックリスト(T B L)は、全てのS A Tブロックのブロックアドレスを規定するため、制御ブロック(後述する)内に保持されている。各S A Tエントリーは、セクタの物理アドレスを規定するとともに、3バイトを占めており、図5に示されるように、

チップ数 5ビット、32個のチップのアドレスが可能。

ブロック数 13ビット、1チップにつき8192個のブロックを設定できる

。

セクタ数 6ビット、1ブロックにつき最大64個のセクタを与えるを備えている。

S A Tページデータ構造上のヘッダーパラメータ領域は、S A Tブロックおよびページ数を含んでいる。

【0063】

8MBおよび8VBの容量のブロックを有するフラッシュカードは約16Kのセクタを記憶することができ、したがって、そのS A Tは約16Kのエントリーを有している。S A T内の512バイトのページは170個のエントリーを有しており、したがって、S A Tは6ブロックに相当する約96ページを占める。大きなフラッシュメモ리카ード(2GB、8VBブロック)は1543個のブロックを占める。

【0064】

追加S A Tブロック(ASB)

追加S A Tブロック(ASB)は、特定のS A TブロックにリンクされてS A Tブロックの1ページを修正する(すなわち、書き換える)ことができる専用のブロックである。幾つかのASBを設けることもできる。各ASBは、それがリンクされるS A Tブロックに対する拡張子として作用する。S A Tブロックが修正される場合、一般に、S A Tブロックは、僅かな数のページ内にだけ修正デー

タを含んでいる。ASBによって、これらの修正ページだけを書き換えることができる。これは、全ページの書き込み、および、SATブロックを書き換えるために必要な廃棄ブロックの消去よりも極めて速い操作である。ASBページのヘッダーパラメータ部分は、それがリンクされるSATブロックとSATページ数とを、それが取って代わるブロック内に含んでいる。ASBページ内に記憶されたデータの情報部分1aのフォーマットは、SATページのそれと同じである。

【0065】

制御ブロック (CBs)

コントローラ8は、1または複数の制御ブロック (CBs) 内に、幾つかの制御アドレス情報を記憶する。CBsは、コントローラ8によって周期的に更新されるとともに、メモリシステム10の初期化中、あるいは、操作中において時々、アクセスされなければならない。情報は、独立に書き込むことができる所定サイズのエントリーの形態で、CBs内に記憶される。各CB内にはページ毎に9個のエントリーが存在する。エントリーは、エントリーそれ自身内のCBヘッダー領域によって認識されるデータタイプの以下のリストのうちの1つに関連している。

- ・テーブルブロックリスト (TBL)
- ・ファイルデータ書き込み操作に対応する、エントリー内に幾らかの領域を有するブロックのマッピング (WMAPI)
- ・システムデータ書き込み操作に対応するブロックのマッピング (SMAP)
- ・再配置セクタ操作に対応するブロックのマッピング (RMAP)

新しいデータをCBに加えなければならない場合には、適当なタイプの別個のエントリーが、最後の有効なエントリーに続くように即座に加えられる。大きなカードにおいては、CBが1ブロック以上を占めていても良い。全てのCBブロックのアドレスは、フラッシュメモリ6内のブートブロック (BB) に記憶される。

【0066】

データのセキュリティのため、CB (及びBBも) の各ブロックの第1ページ (ヘッダーページと呼ばれる) は、エントリーを含んでおらず、図3のような

全ページフォーマットを有している。このページのヘッダー16のヘッダーパラメータ領域は、CBを認識するサインと、そのブロック数（制御ブロックのセット内でのブロックの連続番号）とから成る。

【0067】

第1のCBブロックのヘッダーページの情報領域は、ブロックリンク情報データによって占められる。CB書き換え操作に続いてシステムを初期化しなければならない場合、ブロックリンク情報データは、WBL、WSBL、RBLをリストアするために必要な全ての情報を提供する。ブロックリンク情報データは、最後のSATページ書き込み操作が実行された後に書き込まれた全てのMAP（WMAPs、SMAPs、RMAPs）から収集されたリンク領域を備えており、その書き換え操作中に、新しいCBの第1ブロックのヘッダーページに書き込まなければならない。ブロックリンク情報は、4バイトのエントリーのタイミング順リストであり、各エントリーは、ライトプリンタのうちの1つによってアクセスされるブロックのブロックアドレスと、それがどのポインタであったかを認識するフラグとを含んでいる。メモリ空間内の多数のブロックがCBのために与えられ、これが一杯になると、アクティブエントリーは、NEB（すなわち、それは、コンパクトであり、対応するヘッダーページが加えられる）から、次の利用可能な消去ブロックへと書き換えられる。CBページの情報領域は、長さが等しい9個のエントリーを有している。CBページのEEC領域は、使用されないが、他の目的のために使用されても良い。

【0068】

1つのCB内の1つのエントリーは、56バイトの大きさであり、図6に示されるフォーマットを有しているとともに、エントリーのデータタイプを認識するヘッダー24と、情報領域26と、EEC領域28とを備えている。EECは、全ページのために使用される場合と同様の形態を成している。

【0069】

テーブルブロックリスト（TBL）

CBは、テーブルブロックリスト（TBL）を含んでいる。TBLは、SATの連続するブロックおよび任意の対応するASBからなるブロック群のアドレス

を有している。通常、複数のTBLは、必要な全てのSATブロックアドレスを規定するために要求される。関連するTBLは、修正されたSATもしくはASBのブロック位置を記録するため、SATブロック書き込み操作もしくは新たなASBの割り当ての後、直ちに、書き込まれる。また、TBLは、SATページ書き込み操作が実行される際にその時点でのライトポインタの位置を記録するために使用される領域を有している。したがって、TBLは、SATページ書き込み操作が実行される場合には常に書き込まれる。TBLの1つのエントリーが図7に示されている。このエントリーは、CBエントリーの情報領域26を占めている。1バイトの第1のTBLエントリーは、TBLエントリーの連続番号Noである。各TBLエントリーは、8個のSAT-ASBブロック対の値を維持している。すなわち、各TBLエントリーは、SATブロックN~N+7のための値を維持している。この場合、NはNo*8である。

【0070】

WPはライトポインタページ領域であり、RPはリロケーションポインタ領域であり、SWPはシステムライトポインタ領域である。これらの領域は、WSL、RSL、SSLリリース操作後、ブロック内でのWP、RP、SWPの位置を決定する。WP、RP、SWPは、WSL、RSL、SSLリリース操作が終了した時（すなわち、最後のSAT書き込み操作）だけ有効である。この状態は、エントリーヘッダー内のフラグビットを用いて設定される。フラグ=1は、エントリーがWSLまたはRSLのリリース中に書き込まれた最後の1つであり、したがって、WP領域が有効であることを意味している。

【0071】

保留されている3バイトは、将来の追加の可能性のために残されている。

SAT-ASB対は、保留（Reserved）領域の後に設けられており、8個のエントリーの列である。各エントリーは、SATブロックアドレスとASBブロックアドレスから成る。

SAT_NはN番目のSATブロックの数であり、ASB_NはSAT_NにリンクされるASBの数である。SATもしくはASBが存在しない場合には、この領域の値は0に等しい。

SATブロックが再配置される度に、あるいは、新たなASBがSATブロックにリンクされる度に、新たなTBLエントリーがCBに加えられなければならない。

【0072】

MAP (WMAP, SMAP, RMAP)

CBは、様々なMAPエントリーを含んでいる。3つの異なるタイプのMAPエントリーがある。各マップエントリは、異なるタイプの書き込み操作に対応している。すなわち、WMAPは、ファイルデータの書き込み或は削除操作に対応しており、SMAPは、システムデータの書き込み操作に対応しており、RMAPは、セクタ再配置操作に対応している。

【0073】

全てのMAPエントリーの情報領域は、同じフォーマットを有している。情報領域は、連続ブロック群の消去状態を規定するビットマップを含んでいる。256個のブロックの消去状態は、MAP内の32バイト領域内の256ビットによって規定される。情報領域は、それが関連するブロック群を認識するレンジ (Range) 領域を有している。他の領域は、ブロック間でライトポインタが移動する際の行き先ブロックを規定する。また、MAPは、旧データを含むフラッシュメモリ内のブロックの場所を認識する複数の領域を有している。すなわち、ObsCは、COB (またはCOSB) のために使用され、ObsPは、保留の廃棄ブロックのために使用される。廃棄ブロックが存在しない場合には、対応する領域が0に設定される。EB領域は、現在の書き込み或は削除セクタ操作によって消去が引き起こされる1つのブロックの1つのアドレスを含んでいる。そのようなブロックが存在しない場合には、EB領域が0に設定される。このMAPエントリーフォーマットが図8に示されている。

【0074】

ライトポインタのうちの1つが1つのブロックから他のブロックへと移動される場合には、消去ブロックの使用を示すために (ビットマップ領域が更新される)、また、ブロック間のリンクを記録するために (リンク領域が更新される)、対応するMAPエントリーが加えられなければならない。書き込み (あるいは、

削除セクタ) 操作によって新しいブロック内に廃棄 (または削除) データが形成される場合には、新しい廃棄ブロックの位置を記録するために (ObsCやObsPが更新される)、また、ブロックが消去されるべきであることを示すために (EB領域が更新される)、また、新たな消去ブロックが出現することを示すために (ビットマップ領域が更新される)、対応するMAPも加えられなければならない。したがって、通常、MAPの少なくとも2つの領域が同時に書き込まれ、これは、単一のページ書き込み操作で達成されても良い。

【0075】

ブートブロック (BB)

ブートブロックの機能は、カードからの予期しない電力の除去時にデータの高いセキュリティを提供すると同時に、初期化プロセス中における広範囲な走査を避けることである。大きなカードにおいて、BBは1つ以上のブロックを有している。BBは、カード内で、第1の非欠陥ブロックを占めている。データセキュリティのため、カード内には、次の非欠陥ブロックを占めるBBのコピーが存在する。BB及びそのコピーは、書き換え後、同じ場所に置かれなければならない。

【0076】

BBは、制御ブロックと同じ構造を有しており、以下のタイプのエントリーを含んでいる。

- ・サイン
- ・インターリーピングイネーブル
- ・不良ブロックリスト
- ・制御ブロックポインタテーブル (CBPT)

新しいデータをBBに加える必要がある場合には、最後の有効エントリーの後に直ぐ続くように、適当なタイプの別個のエントリーが加えられる。サインとBBLEエントリーは、前述したものと全く同じフォーマットを有している (無論、サインエントリーのサイン領域は、異なっており、固有のものである)。制御ブロックポインタテーブルエントリーは、CBの全てのブロックに対するポインタを有しており、CBが書き換えられた直後に更新される。

【0077】

予備リンク (PL)

また、CBは、予備リンク (PL) を有している。予備リンクの目的は、CB書き換え操作後にシステムを初期化する必要がある場合に、WBL (後述する) をリストアするために必要な全ての情報を提供することである。予備リンクは、最後のSATページ書き込み操作が実行された後に書き込まれた全てのMAPから収集されたリンクファイルを備えている。PLは、その書き換え操作中、新たなCBに対してだけ書き込まれる。

【0078】

コントローラのSRAM内に記憶されたデータ構造

ライトセクタリスト (WSL) (すなわち、ホイッスル) を含むマイクロプロセッサのSRAM13内には、様々なデータ構造が記憶されている。ライトセクタリストは、最後のSAT書き込み後に書き込まれたセクタの論理アドレスを記録する。その目的は、そのようなセクタのために、論理アドレスから物理アドレスへの正確な変換を与えることである。WSLは、4バイトの長さの128個のエントリーのための容量を有している。この場合、各エントリーは、連続的に書き込まれるセクタ群内の第1のセクタの論理アドレスと、セクタ群の長さを記憶する。セクタ群は、1つのブロックから他のブロックへとジャンプすることができない。WSLは、SAT書き込み直後に空になる。

【0079】

WSL内の論理セクタアドレスの順序は、それらが書き込まれる順序と正確に一致する。したがって、フラッシュメモリ内のWSLの不揮発性コピーは、SATが最後に書き込まれた時にライトポインタ (WP) によって規定される場所で始まる連続的な場所で書き込まれる実際のセクタのヘッダーによって自動的に提供される。したがって、WSLのコピーをフラッシュメモリ6に明確に形作る必要はない。必要に応じて、これらのセクタは、次のMAPエントリーからのリンク領域の他、SATが最後に書き込まれた時のライトポインタ (WP) の位置を含む制御ブロック領域内に規定される開始アドレスから走査することができる。

【0080】

任意の論理セクタのための最後のエントリーだけが有効であるため、WSLの検索は、逆の順序で実行される。セクタの再配置およびブロックの消去が行なわれた場合があるため、先の複製エントリーは、フラッシュメモリ内に置かれた任意の対応する廃棄セクタを有していなくても良い。好ましい場合には、WSLから複製エントリーを単に除去するようにコントローラが構成される。

【0081】

また、2つの類似するリスト、すなわち、再配置セクタリスト (RSL) およびシステムセクタリスト (SSL) は、最後のSAT書き込み後に書き込まれるシステムデータセクタ (RPおよびSWPのそれぞれによって指し示されたアドレスに書き込まれる) および再配置されたセクタの論理アドレスを記録するマイクロプロセッサのSRAM13内でコンパイルされる。

【0082】

WSL、RSL、SSLがそれぞれ一杯になる度に、ASB及び／またはSATブロックは、WSL、RSL、SSLエントリーによって補われる。このような手続きは、WSL、RSL、SSLリリースと呼ばれる。このリリースは、必要に応じて、ASBリリースを引き起こす。ASBリリースは、ASBが一杯になった時に生じる。ASBが一杯になると、各SATブロックが書き換えられ、ASBが消去される。全てのASB内に書き込まれたページに関する情報は、ASB走査を頻繁に行なわないで済むように、RAM内に記憶される必要がある。そのため、ASBリスト (ASBL) は、SRAM13内に記憶される。

【0083】

ASBLは、SATブロックと現時点でリンクされている全てのASBブロックのリストであり、ASBL内には各ASB毎に1つのエントリーが存在する。図9は、ASBL内の1つのエントリーのフォーマットを示している。この場合、LWP=このASBブロック内に最後に書き込まれたページの数であり、NVP=ASB内の有効ページ数-1である。ASBページ0... ASBページn=1つの列、指数はASBページ数、値は対応するSATページ数である。N=1ブロック当たりのページ。

【0084】

ライトブロックリスト (WBL)

ライトブロックリストは、ライトセクタリストを補って、マイクロプロセッサのSRAM13内に形成され、WSL内のセクタが配置されるブロックを規定する。WBLは、WSLリリース直後に空になる。

【0085】

WSLおよびWBLが、メモリシステムの初期化中に、走査プロセスによって再構成される。このようにして構成されるSRAM内のリストは、電力が最後に除去される前に存在したリストと正確に一致する。

【0086】

また、再配置ブロックリスト (RBL) およびシステムブロックリスト (SBL) と称されるWBLに類似する2つのリストは、コンパイルされて、SRAM13内に記憶される。これらのリストは、RSLおよびSSLをそれぞれ補う。RBLおよびSBLは、RSLおよびSSL内のセクタがそれぞれ物理的に配置されるブロックを規定するとともに、WBLと類似するフォーマットから成る。また、RSL、RBL、SSL、SBLは、メモリシステムの初期化中に、走査プロセスによって再構成することができる。

【0087】カレントオブソリートブロック (COB)

ライトポインタ (WP) によって書き込まれる廃棄もしくは削除セクタデータを含むブロックは、たった1つだけその存在が許される。これは、カレントオブソリートブロック (COB) と称される。廃棄もしくは削除セクタデータが他のブロック内で形成される場合、消去操作は、COBとして規定されたブロック上で直ちに実行されなければならない。COBの現在のブロックアドレス、および、このブロック内で廃棄となる或は削除されたセクタのリストは、以下にCOB構造と称するデータ構造として、マイクロプロセッサのRAM13内に記憶される。COB構造は、初期化中において、廃棄ブロックのアドレスを含む領域を最新のMAPエントリからコピーするとともに、再構成されたWSLおよびWBLが解析された後に廃棄セクタアドレスを加え、削除されたセクタアドレスをこのブロックから読み込むことによって、確立される。新たな削除セクタ操作が実

行される度に、あるいは、新たな廃棄セクタが形成される度に、COB構造が更新され、また、旧データが新たなブロック内で形成される度に、CB内の現在のMAPエントリーにブロックアドレスがコピーされる。

【0088】

SWPによって書き込まれる廃棄もしくは削除セクタデータを含む1つのブロックの存在もまた許される。このブロックは、カレントオブソリートシステムブロック(COSB)と称される。また、COSB構造は、COB構造と類似する方法でSRAM13内に記憶される。COBおよびCOSBはそれぞれ、図10に示されるフォーマット、すなわち、4バイトのブロック数領域28(これは、COBまたはCOSBのブロックアドレスである)と、このブロックの内側の廃棄もしくは削除セクタと対応する位置にマスクを有するビットマップである32バイトの廃棄もしくは削除セクタマスク30とを有している。256ページから成るブロックにおいて、このマスク30は32バイトを要する。

【0089】

次消去ブロックリスト(NEB)

ブロック間でWP、SWP、RPを増加する際に次に利用可能な消去ブロックを早期に認識するため、次消去ブロックリストは、マイクロプロセッサのSRAM13内に形成される。利用可能な消去ブロックは、その物理アドレスの順位を上げる際に使用される。NEBは、M個の消去ブロックアドレス(例えば、M=8)を有している。NEBリストは、次に利用可能な消去ブロックのリストであり、使用のために割当てられる最後の消去ブロックのアドレスに最も近く且つそれよりも高いブロックアドレスを有する消去ブロックで始まる。したがって、NEB内のエントリーの数には制限される(例えば8個に制限される)が、NEBそれ自身は、次の8個の消去ブロックよりも多い情報を含んでいても良い。

【0090】

次消去ブロックリストは、ライトポインタによってアクセスされるフラッシュメモリのアドレス空間の領域に適したMAPエントリー(CB内に記憶される)から得られる。次消去ブロックリストは、これによって規定される全ての消去ブロックが使用されるまで、アクティブNEBとしてSRAM内に残される。そし

て、全ての消去ブロックが使用された時点で、次消去ブロックリストは、適当なMAPエントリーから再構成されなければならない。CBは、フラッシュメモリ内の全てのブロックの消去状態を規定できる十分な量のMAPエントリーを有している。NEBおよび対応するMAPエントリーは、メモリシステムの操作中にブロックを加減することによって更新される。セクタもしくは制御データ記憶装置のために消去ブロックが割当てられた時に、エントリーがNEBから除去される。NEB内の他のブロックによって測られる領域内にあるブロックアドレスに消去ブロックが形成されると、エントリーがNEBに加えられる（ただし、NEBが既に一杯である場合を除く）。

【0091】

単一のNEBエントリーは、連続するアドレスで消去ブロック群を規定するとともに、ブロック開始アドレス（連続するブロック群内の第1のブロックアドレス）および群の長さ（群の連続するブロックの数）を規定する。

【0092】

TBLポインタ (TBLP)

TBLポインタは、CB内のTBLエントリーの位置を認識する。また、TBLポインタは、マイクロプロセッサのSRAM内に記憶されるとともに、高速セクタアドレス変換を提供するために使用される。TBLPは、制御ブロックの初期走査中に形成され、その後、新たなTBLエントリーがCB内に形成される度に更新される。

【0093】

ASBリスト (ASBL)

前述したように、ASBLは、SRAM13内に形成されるとともに、高速アドレス変換をサポートする。ASBLは、全てのASBブロック及びこれらのASBブロックが関連付けられるSATブロックの物理アドレスを認識する。ASBLは、初期化中に形成され、SATページ書き込み操作が実行される度に更新されなければならない。ASBLエントリーは、以前のアクセスの順序でリストアップされる。すなわち、エントリーは、アクセスされる際、リストのトップに起用される。対応するASBを現時点で有していないSATブロックがアクセス

されると、ASBが割当てられ、そのためのエントリーがASBLのトップに加えられる。最も古くにアクセスされたASBを示すASBLの下位のエントリーは削除される。

【0094】

SATキャッシュ

キャッシュは、フラッシュメモリ内のSATから最も新しくアクセスされたエントリーを組み込む32個の連続するSATエントリーのため、SRAM13内に維持される。

【0095】

キャパシティーマップ

メモリスistem10内のフラッシュメモリの総容量は、以下のように、データおよび制御構造に割当てられる。

1. 論理セクタ

容量は、カードの定格論理容量内で、各論理アドレスのための1つの有効データセクタの記憶に割当てられる。この定格容量は、利用可能な物理容量からアイテム2～8を引いたものよりも小さく、カード製造中においてフォーマッターにより規定される。

【0096】

2. ブートブロック

少なくとも1つのブロックがブートブロックに割当てられる。好ましくは、ブートブロックの他のコピーを記憶するために、第2のブロックが割当てられる。

【0097】

3. 制御ブロック

制御ブロックエントリー（サイン、BBL、TBL、MAP）の記憶に多数のブロックが割当てられる。十分にコンパクトな制御ブロックは、大抵の場合、占有面積が1ブロックよりも小さい。圧縮／書き換え操作間で書き込まれるエントリーのための制御ブロックに別個のブロックが割当てられる。

【0098】

4. セクタアドレステーブル

これは、SATのブロックの記憶に割当てられる容量である。この容量は、カードの論理容量に比例している。

【0099】

5. 追加SATブロック

所定のSATブロックに関連付けられるASBのために、所定数のブロックが割当てられる。

【0100】

6. 廃棄セクタ

COBのために1つのブロックが割当てられる。COSBのために他のブロックが割当てられ、POB（COSBおよびPOBの存在を許容する実施例において）のために更に1つのブロックが割当てられる。したがって、許容される旧データセクタの最大数は、ブロック内のページ数によって設定される。

【0101】

7. 消去バッファ

これは、システムの修正操作のために割当てられなければならない消去ブロックのバッファである。データセクタ再配置のために、少なくとも1つの消去ブロックが割当てられなければならない、同時に生じる可能性がある制御構造再配置のために1つの消去ブロックが割当てられなければならない。

【0102】

8. スペアブロック

スペアブロックは、作動期間中の故障時に定格論理容量を維持するために使用されるべく割り当てられる。スペアブロックの数は、カード製造中においてフォーマッターにより決定される。

【0103】

8MBカード、64MBカード、512MBカード（フラッシュカード）に関し、例えば前述した項目1～8に割当てられる容量を示す容量割り当てテーブルが図16（これは、廃棄セクタのため、すなわち、COBのために割当てられる1つのブロックだけを示しているが、無論、これよりも多くのブロックが廃棄セクタのために割当てられても良い。その場合には、メモリシステム内のCOSB

やPOBのための対策が必要となる)に示されている。

【0104】

SAT書き込み操作

SATは、WSL、WBL、SSL、SBL、RSL、RBLが一杯になった時に書き換えられる。大抵の場合、それは、1ページの粒度をもって行なわれる。すなわち、修正されたセクタアドレスを含むSATページだけが書き換えられる。しかしながら、極稀に、SATブロック全体の書き換えが必要な場合がある(「SATブロック書き込み」と呼ばれる)。SAT書き換えが必要とされる場合には、追加SATブロック(ASB)が形成され、その中に、必要とされるページだけが書き込まれる(「SATページ書き込み」と呼ばれる)。特定のSATブロックに専用の1つのASBがあっても良く、また、全体として、ASBの制限された数Nが存在していても良い。Nの値は、書き込み特性と容量オーバーヘッドとの間の適当な妥協点として選択される(例えば、N=8)。

【0105】

各ASBは、「SATブロック書き込み」がその対応するSATブロック上で実行されるまで存在する。ASBで一杯となるSATブロック上もしくはASBの割り当てが取り消されるSATブロック上でSAT書き込みページが必要とされる時に、SATブロック書き込みが実行される。SATページ書き込みが必要とされ、対応するSATブロックが関連するASBを有しておらず、また、N個の全てのASBが既に割り当てられている場合には、新たなASBの割り当てが成される前に、既存のASBのうちの1つの割り当てが取り消されなければならない。割り当てが取り消されるASBは、最も長い時間の間書き込まれなかったASBとして選択される。ASBの割り当ての取り消しは、SATブロックの書き込みと廃棄SATブロックおよびASBの消去とを必要とするため、幾分時間を消費する手続きである。

【0106】

インターリーブチップアクセス

前述した操作およびデータ構造によって、本質的に、複数のフラッシュチップ5上でインターリーブ(交互配置)書き込み操作を実行することができる。これ

により、性能を十分に向上させることができる。したがって、コントローラチップ8は、複数のフラッシュチップ5、例えばメモリシステムに組み込まれる4個のフラッシュチップの列を制御しても良い。4個のフラッシュチップのメモリ空間は、コントローラにより、仮想ブロックのセットとして編成される。各仮想ブロックは4つのブロックから成り、1ブロックは4つのチップ5のそれぞれからのものである（チップは共に永久にリンクされる）。各仮想ブロック内の4つのブロックは、1チップ内に同じブロックアドレスを有するブロックである。このような編成によれば、仮想ブロックを形成するリンクされた複数のブロックを1つの大きなブロックとして処理することができ、したがって、セクタの個々の消去ブロックの代わりに仮想ブロックを使用して、単一のフラッシュメモリシステム10において述べたと同様のアルゴリズムおよびデータ構造の全てを使用することができる。ページ書き込み操作は、全てのインターリーブチップにわたって同時に実行される。ライトポインタ（WP）およびRP、SWPはそれぞれ、連続的にページアドレスを進め、コントローラ8内に設けられたハードウェアチップイネーブルデコーダに供給されるアドレスビットの順序により、リンクされたブロックのページは、図11に示される順序で連続的にアクセスされる。すなわち、各ポインタWP、SWP、RPは、1つのPAから他のPAに移動する際、1つのチップから他のチップへと移動する。これは、仮想アドレスを使用することによって達成される。固有の仮想アドレス（VA）は、全てのチップ内の各物理セクタに割当てられる（例えば、前述したNANDを基本とするメモリシステムにおいて、VAは各ページのために割当てられる）。1つだけVAを増加させることによってライトポインタが1つのチップから他のチップへと移動するように、仮想アドレスが割当てられる。仮想アドレスは、各仮想ブロックのリンクされたブロックを介して、図11に示されるようなパターンで、チップからチップへと増大する。

【0107】

実際、コントローラは、メモリーチップアレーを仮想ブロックの単一の縦列として処理する。セクタの仮想アドレスは、図12に示されるフォーマットを成している。これは、チップハイ（ChipHigh）部分およびセクタの13ビットのプロ

ックアドレスを備えた仮想ブロック部と、6ビットのページアドレスおよびチップロウ (ChipLow) 部分を備えた仮想ページ部とから成る。チップハイ部分は、C h i g h ビットの5ビットチップ数であり、チップロウ部分は、C l o w ビットの5ビットチップ数である。この場合、

C h i g h = チップアレー内のチップの縦列数、

C l o w = チップアレー内の横列数である。

仮想アドレス (VA) から物理アドレス (PA) を得るため、コントローラは、図13に示されるように、チップハイ部分とブロックアドレス部分との間にチップロウ部分を逆行させるようにVAを単に再編成する。したがって、任意のセクタのための単一のチップメモリシステムにおいて、VAはPAと等しい。

【0108】

簡素化するため、2倍数のチップだけ、例えば2個と4個のチップをインターリーブ (交互配置) しても良い。仮想ブロックにおける消去操作は、インターリーブされたチップのリンクされた全てのブロックの同時消去として実行される。1つのチップ内の1つのブロックが不良ブロックである場合、コントローラは、等しいブロックアドレスを有する他のチップの全てのブロックを不良ブロックとして処理する。

【0109】

フラッシュメモリがフォーマットされる時にメモリシステムの製造メーカによって可能状態もしくは不能状態に設定されるブートブロックの制御バイト状態によって、インターリーブングが可能であったり、不能であったりする。

【0110】

前述した単一のチップNAND型フラッシュの実施例においてブロックアドレスすなわちPA s が予め使用される場合に、仮想ブロックアドレスおよびVA s をそれぞれ使用する。コントローラは、ホストデータセクタ書き込みコマンドを受け取ると、関連するライトポインタが指し示しているPAを割当てることによって、入ってくるLAをPAに変換する。コントローラは、複数のライトポインタを制御して、各ライトポインタをPAを通じて移動させる。その結果、ライトポインタは、消去される仮想ブロック (消去された仮想ブロックはNEBで認識

される)のそのセクタの仮想アドレス(VAs)を通じて、連続的に移動する。

【0111】

図14は、インターリーブチップへの多数のセクタ書き込みに関与する様々な操作のタイミングを示している。以下、図15および図19を参照しながら図14について説明する。図15は、メモリシステムのコントローラチップ8を詳細に示すブロック図である(図2のコントローラチップが図15に示される形態を成していても良く、図2および図15の両者において同じ構成要素が同一の符号で示されている)。図19は、コントローラチップ81と4つのフラッシュチップ1', 2', 3', 4'とを備えたメモリシステム10の概略図である。各フラッシュチップは、それ自身のリード/ライトバッファ71', 72', 73', 74'を有している。

【0112】

図15は、ホストコンピュータとの接続のための入/出力ポートO/P、例えば、PCカードATAポート、コンパクトフラッシュ(登録商標)またはIDEポートと、O/PおよびデュアルポートSRAMセクタバッファ9に接続されたホストインタフェース&レジスタ80と、データバスコントローラ82と、ECCジェネレータ&チェッカ12と、フラッシュメモリインタフェース(FMI)84とを有するコントローラチップ8'を示している。データバスコントローラ82と、ECCジェネレータ&チェッカ12と、フラッシュメモリインタフェース(FMI)84は全て、セクタバッファ9に接続されており、また、FMIはフラッシュメモリポート86に接続されている。また、コントローラ8は、マイクロプロセッサ11(RISCプロセッサの形態を成す)と、プロセッサSRAM13と、プロセッサマスクROM88と、外部プログラムRAMまたはROM92のためのポートとを有している。また、RISCプロセッサ11のため、オプションで、デバッグポート94が設けられていても良い。データおよびコマンドは、マイクロプロセッサバス91を介して、コントローラ8'(セクタバッファ9を除く)の様々な構成要素間で通信される。

【0113】

図14に示されるように、複数のセクタ書き込みコマンド(この場合、ホスト

データセクタ1, 2, 3, 4から成る4つのセクタ書き込み)がATAポートO/Pで受けられると、デュアルポートセクタバッファ9のバッファ1にセクタ1が書き込まれる。これにより、コントローラのデータ管理操作のために利用可能なセクタバッファ9のバッファ2が残される。セクタ2が受けられると、セクタ2はバッファ2に直接に書き込まれ、同時に、セクタ1がバッファ1からフラッシュメモリポート86に移動される。そして、セクタ1は、このフラッシュメモリポート86から、4つのフラッシュチップのうちの1つ(チップ1')のリード/ライトバッファ71'へと書き込まれる。その後、セクタ2は、バッファ2からフラッシュポート86へと送られ、そこから、4つのフラッシュチップの他の1つ(チップ2')のリード/ライトバッファ72'へと書き込まれる。このようなことが起きている間、セクタ3がセクタバッファ9のバッファ1に直接に受けられる。セクタ3はチップ3のバッファ73'に書き込まれ、セクタ4は、セクタバッファ9のバッファ2に受けられた後、チップ4のバッファ74'に書き込まれる。その後、セクタ1, 2, 3, 4は、チップ1', 2', 3', 4'のメモリアレー61', 62', 63', 64'内の関連する割り当てられた物理セクタにそれぞれ書き込まれる。図14は、全ての意図および目的のため、そのようなセクタ書き込み操作のそれぞれが前のセクタ書き込み操作の直後に開始される形態を示しているが、無論、4つのセクタ1, 2, 3, 4が略同時にフラッシュチップ1'~4'に書き込まれても良い。また、ホストデータセクタ1~4が書き込まれるセクタの物理アドレスは、関連するライトポインタの位置を決定する前述したアルゴリズムによって決定されても良い(すなわち、仮想アドレスの連続使用)。

【0114】

4つ以上のセクタを書き込む多重セクタ書き込みコマンドが、ホストプロセッサからコントローラに送られ、コントローラが多重セクタ書き込みを4つのセクタの群(この例では、4つのメモリチップを使用する)に分割するようになっていても良い。この場合、そのような各群は、図14において前述したように、インターリーブ書き込みシーケンスで、フラッシュメモリに書き込まれる。

【0115】

アドレス変換

以下、読み込み及び書き込み操作に関して、LAをVAに変換するアドレス変換プロセスを詳述する。

【0116】

アドレス変換は、以下の3つの値のうちの1つを戻すセクタの論理アドレス上で実行される操作である。

- ・有効セクタ物理アドレス
- ・論理セクタが削除された（あるいは、論理セクタが決して書き込まれない）情報
- ・エラー状態が生じた情報

【0117】

図17は、アドレス変換プロセスを示すフローチャートである。このプロセスは、全ての読み込み操作において実行される。変換される論理セクタアドレス（LA）がホストプロセッサからコントローラ8'によって受けられると、アルゴリズムが実行され（ボックス40）、変換される物理アドレスを有するセクタが、そのSATエントリーの最後の書き込み後に、予め、書き込まれ或は削除された可能性があるか否かが認識される。セクタが予め書き込まれ或は削除されたという結論は、100%の確実性を有していなければならない。すなわち、セクタが予め書き込まれ或は削除されたという結論は、高い確率で正しくなければならない。これは、連続するセクタ系列の開始アドレスと終了アドレスとを示す値の対の限られた数を維持することによって、また、変換されるアドレスがこれらの任意の範囲内にあるか否かを認識することによって、実行される。これらの範囲が最終的に非連続となって、セクタが予め書き込まれ或は削除されたという結論が不確実になっても良い。LAが任意の範囲内にある場合、「繰り返しの可能性は？」の問いに対してYESで答える（図17のボックス42）。LAが任意の範囲内にない場合には、NOと答え、VAを見つけるために、SATまたはSATキャッシュへと進む（ボックス44）。ここから、物理セクタが不良であるか否か（46）、あるいは、削除されたか否か（50）を決定する。LAが書き込まれないセクタに対応している場合には、ボックス50でVA=削除（48）と

なる。ボックス42でYESと答えると、ボックス52で、WSLもしくはSSL (LAがファイルに対応しているか或はシステムデータに対応しているかどうかに応じて)を検索する。ボックス54で、LAが見つかる (YES) と、VAを演算し (56)、物理セクタのヘッダー1b内に記憶された論理アドレスが、コントローラマイクロプロセッサによって読み込まれる (58)。LA=LA1である場合 (ボックス60) には、演算されたVAは正しい。LAが54で見つからない場合には、RSL内でLAを検索し (62)、LAがRSL内で見つからない場合には、SATまたはSATキャッシュへと進んで、そこからVAを得る (44)。SATもしくはSATキャッシュ内で見つかったVAが不良もしくは削除されている場合 (46, 50) には、前述したように、VAからLA1を得て (58)、LA=LA1であるかどうかをチェックする。

【0118】

図17のボックス56で実行されるプロセスステップ (VAの演算) が図20のフローチャートに詳細に示されている。図20は、たった2つのライトポインタWP, RPを使用するメモリシステムにおいて、WSLまたはRSL内で見出されるLAのためのVAを得るために実行されるステップを示している。無論、このフローチャートは、メモリシステムがシステムライトポインタ (SWP) をも組み込んでいる場合において、SSL内で見出されるLAのためのVAを与えることができるように拡張されていても良い。プロセスはNumFromEnd (NFE) を設定するボックス110で開始される。NumFromEndは、WSL (またはRSL) の終端から書き込まれる所定のセクタ (WSLまたはRSL内に見出されるセクタ) までのセクタ数である。LAがWSL内で見つけられた場合には、P=WPに設定し、LAがRSL内で見つけられた場合には、P=RPに設定する。そして、その後、PG=P. Pageに設定する。この場合、P. Pageは、Pの値によって表わされるライトポインタのページのことである (ボックス102参照)。104において、PG>NFEである (すなわち、LAが最後に書き込まれたブロック内にある) 場合には、VA=P-NFE-1、すなわち、NFE-1のセクタが関連するライトポインタの位置から離れる。PG<NFEである場合には、106で、P==0であるか否か、すなわち、最後のWBL/RBL

エントリーに対応するブロックが十分に書き込まれたか否かが決定される。書き込まれている場合（すなわち、 $P=0$ ）には、NotLastが0に設定され、書き込まれていない場合には、NotLastが1に設定される。その後、108において、与えられたセクタが存在するブロックと最後のブロックとの間のブロック数Nbblockが、以下のアルゴリズムを使用して演算される。

Nsectは、与えられたセクタと最後に書き込まれたブロックページ0との間のセクタ数である。

$$Nsect = NumFromEnd - PG$$

$$Nbblock = Nsect / BlockSize + NotLast$$

【0119】

その後、110において、ブロック内のページ数PageNumを演算する。この場合、 $PageNum = BlockSize - Nsect \% BlockSize$ である。その後、以下の手法を使用して、LAがWSL内にある場合には、112で、WBLからブロックアドレス(BLAddr)を取得し、LAがRSL内にある場合には、114で、RBLからブロックアドレス(BLAddr)を取得する。この場合、ブロックアドレスは、所定のセクタを含むブロックの仮想アドレスである。

LAがWSL内にある場合には、 $BLAddr = RBL [LBL - Nbblock]$ 、この場合、LBLはWBL内の最後のエントリーのインデックス；

LAがRSL内にある場合には、 $BLAddr = RBL [LRBL + Nbblock]$ 、この場合、LRBLはRBL内の最後のエントリーのインデックスである。

【0120】

その後、116で、 $VA = Page0 + PageNum$ を使用して、VAを演算する。この場合、Page0は、与えられたセクタを含むブロック内のページ0の仮想アドレスである。

【0121】

図21は、SATまたはSATキャッシュからVAを取得する（図17のボックス44）ために使用されるプロセスのフローチャートである。図21は、一般に自明であるが、以下、特に番号が付与されているボックスについて説明する。

【0122】

ボックス120 (LAがSATキャッシュ内にあるか) : $LA \geq \text{FirstCacheEntry}$ である場合には、LAがSATキャッシュ内にある。 $LA < \text{FirstCacheEntry}$ 、 $LA + \text{CacheSize}$ 、この場合、 FirstCacheEntry 。LAは、キャッシュ内の最初のSATエントリーに対応するLAであり、(全体の) CacheSizeは、SATキャッシュ内のエントリーの数である。

【0123】

ボックス122 (SAT内のブロックおよびページを演算する) : ここでは、与えられたLAにおけるSATブロック数であるSBNuと、与えられたLAにおけるSATページ数であるSpaとを演算する。

【0124】

ボックス124 (TBLエントリーの数を演算する) : TBLNuは、要求されるTBLエントリーの数である。この場合、 $\text{TBLNu} = \text{SBNu} / 8$ である。

【0125】

ボックス126 (SATページの一部をキャッシュ内に記憶する) : 可能であれば、最後にアクセスされたエントリーで始まる32個のエントリーがキャッシュされる。その後、十分なエントリーが無い場合には、ページ内の最後のエントリーで終わり且つ最後のセクタアクセスを含む32個のエントリーから成る1つの群がキャッシュされる。

【0126】

図18は、図17のボックス58 (VAからLA1を取得) のためのプロセスステップを示すフローチャートである。ページヘッダー内に記憶されたヘッダーパラメータ (HP) は、論理アドレス (LA) の1だけ増大された値である。これは、削除されたセクタが、そのヘッダー内の全てのビットを0に設定することによってマークされているからである。この $LA = 0$ をヘッダー内に記憶することはできない。したがって、 $LA1 = HP - 1$ に設定する。

【0127】

読み込み操作

図22は、物理セクタからホストデータセクタを読み込むために実行されるステップのシーケンスを示すフローチャートである。コントローラは、LA (ホス

トから受ける)をVAに変換することによってスタートする(ボックス130)。これは、既に説明した図17のフローチャートに示されたプロセスを実行することによって成される。LAがVAに変換されると、直ちに、アドレスVAを有する物理セクタの内容が、コントローラのバッファ9に読み込まれる。その後、コントローラは、セクタ(すなわち、セクタの内容)が削除されたか否か或は不良であるか否かをチェックする(ボックス132)。セクタが削除され或は不良である場合、コントローラは、ホストプロセッサのデータバッファの全てのバイトを0xFFに設定する(ボックス134)。セクタが削除されたセクタでない場合には(ボックス136)、コントローラは、エラー状態をホストに戻す(ボックス138)。セクタが削除されたセクタである場合には、コントローラは、有効状態をホストに戻す(ボックス137)。ボックス132において、セクタが削除或は不良でないことをコントローラが決定した場合には、コントローラは、そのままボックス137に進む。すなわち、有効状態をホストに戻す。

【0128】

書き込み操作

図23は、ホストデータセクタを物理セクタに書き込むために実行されるステップのシーケンスを示すフローチャートである。図23は、ホストファイルデータの書き込み操作、すなわち、ライトポインタ(WP)による書き込みだけを扱う。しかしながら、メモリシステムがシステムデータのための別個のライトポインタ(すなわち、SWP)を使用する場合には、図23の操作を適当に拡張して、別個のシステムデータの書き込みを扱えるようにしても良い。コントローラは、LA(ホストから受けられる)をVAに変換することによってスタートする(ボックス150)。これは、既に説明した図17のフローチャートに示されるプロセスを実行することによって成される。152で、セクタが不良である場合には、コントローラは、エラー状態をホストに戻す(154)。セクタが不良でない場合には、セクタが削除されるか否かをチェックする(156)し、セクタが削除される場合には、その後、WPが有効か無効かをチェックする(158)。全てのブロックが書き込まれてしまっている場合には、WPは無効(WP==0)であり、WPを消去ブロックに移動させなければならない。WPが有効でない

場合、WPを新しい（有効）物理セクタアドレスに設定する（160）。WPが有効である場合には、WSLにLAを加え（162）、任意のWSLもしくはRSLリリースや、必要なCBおよびCBPT圧縮を実行する。その後、繰り返し可能性評価アルゴリズム（図17のボックス40）から範囲を更新し（164）、コントローラバッファからWPのアドレスにセクタを書き込み（166）、有効状態の値をホストに戻す（168）。ボックス156で、セクタが削除されないことが分かったと、COB内にVAがあるか否かをチェックする（157）（VAがVBと一致している場合、VAはCOB内にある。この場合、VBは、SRAM13内に記憶されたCOB構造内の仮想ブロック数領域（これは、仮想ブロックアドレス…図12参照）である）。VAがCOB内にある場合には、コントローラのSRAM13内に記憶されたCOB構造内にVAを廃棄として記録し（159）（これは、SRAM13内のCOB構造のビットマスク領域内で、対応するビットを1に設定することにより成される）、その後、ボックス158に進む（WPは有効か）。VAがCOB内にはない場合には、COBを変更し（161）、その後、ボックス159に進む（VAを廃棄としてマークする）。

【0129】

図24は、図23のボックス161で実行されるステップ（COBの変更）のフローチャートである。図24は、一般に自明であるが、以下の注記と併せて読まなくてはならない。すなわち、

ボックス200（VA、B1、VB）：VA、B1はVAの仮想ブロック領域であり、VBは先に説明したものと同一である。

ボックス202（COBは無効か）：VB=0の場合、COBは無効である。VBが0である場合、これは、この時点で旧データが無いことを示している。

ボックス203、204（MaxRelの演算）：MaxRelは、COBから再配置されるセクタの最大数である。MaxRel=P.Page-1である。この場合、P.Pageは、WPまたはRPのページ領域（アドレス）である。

ボックス205、206（ダミーエントリをWSLに加える）：再配置されるブロックがまだ一杯に書き込まれていない場合には、対応する「ダミー」のセクタLAを最後のWSL（RSL）エントリに加えないといけない。

ボックス207（セクタの再配置）：図25を参照。

ボックス208（WMA Pの書き込み）：WMA PをCBに書き込む。この場合、EB+VBおよびビットマップ内の対応するビットは1に設定される。必要に応じて、CBの書き換えを実行する。

ボックス209（リストの更新）：VBと等しいWRBArrayエントリーを見つけ、それを無効としてマークするとともに、WRBArray内の同じVBに関する任意の他のエントリーを無効としてマークする。WRBArrayは、実際には、最初にカウントアップするWBLエントリーと最後にカウントダウンするRBLエントリーとともに、メモリの同じ領域内に実際に記憶されるWBLおよびWSLリストである。2つのリストが中央で交わると、WRBArrayが一杯になる。

ボックス210（COBのセットアップ）：COB構造をSRAM内で更新する。VB領域はVA. B1に設定される。VA. Pageに対応する廃棄および削除マスク（Obs and Del Mask）ビットは1に設定される。他の全てのビットは0に設定される。

【0130】

図25は、図24のボックス207で実行されるステップ（セクタの再配置）を示している。これは、以下の注記と併せて読まなくてはならない。すなわち、

ボックス220, 222, 230：SRAM内のCOB構造の廃棄および削除マスク領域を通り抜けるループを実行する。

ボックス223（セクタは有効か）：ODMask[i] = 0である場合、セクタは有効である。すなわち、COBの廃棄および削除マスクにおける0値は、このページが有効セクタを含んでいることを示している。

ボックス224, 225：RPによって指し示されるブロックが既に一杯に書き込まれている場合には、RP = 0である。

ボックス226（ページヘッダーからのLAを記憶する）：ページヘッダーから得られたLAは、一時的に記憶されて、エントリーをRSLに加える際に使用される。

ボックス225（RP = 0）：エントリーをRSLに加える。必要に応じて、WSL/RSLリリースを実行する。

【0131】

図26は、図23のボックス160で、ライトポインタ(WP)を設定するために実行されるステップを示している。図25のボックス227でRPを設定するために、同様のプロセスが使用される。図26は、以下の注記と併せて読まなくてはならない。すなわち、

ボックス240(WRArrayが一杯ではないか) : $Last < LastRE - 1$ である場合、WRArrayは一杯ではない。ここで、Last(全体的)は最後のWSLエントリーのインデックスである。LastREは、最後のRSLエントリーのインデックスである。

ボックス242(WRBlockが一杯ではないか) : $LBL < LRL - 1$ である場合、WRBlockは一杯ではない。ここで、LBLおよびLRLは、図20において先に定義したものと同様である。

ボックス244(リストのリリース) : SATページ書き込み操作を実行する。必要であれば、SATブロック書き込みおよびCB書き込みを実行する。

ボックス246(NEBを満たす) : CB内に記憶されたMAPから次のN($N = NEBSize$)の消去ブロックを選択する。

ボックス248(WMAPの書き込み) : ErBlockに設定されたリンク領域および0に設定されたビットマップ領域内の対応ビットとともに、WMAPを書き込む。必要に応じて、CB書き換えを実行する。

【0132】

図27は、削除セクタ操作において実行されるプロセスステップのフローチャートである。これは、初期のステップにおいては、書き込み操作(図23参照)に類似している。すなわち、LAがVAに変換され(250)される。252で、セクタが削除される場合には、エラー状態をホストに戻す(254)。セクタが削除されない場合には、VAがCOB内にあるか否かをチェックして(257)、VAがCOB内に無い場合には、COBを変更する(261)(図24と同様)。VAがCOB内にある場合には、COB内でVAを削除としてマークし(269)、その後、デュアルポートSRAM9(コントローラ内)のバッファのうちの1つを複数の0で一杯にする(271)。その後、バッファからのこの「

オールゼロ」ページを、削除されるセクタに書きこむ（これによって、セクタを削除する）。その後、有効状態をホストに戻す（275）（セクタが削除されたことを確認する）。

【0133】

初期化

以下、初期化手続き及び電力損失リカバリ手続きについて説明する。簡単のため、単一のライトポインタシステム（すなわち、WPだけ）に関してこれらを説明する。しかしながら、無論、マルチプルライトポインタシステム（WP、SWP、RP）に適するように手続きを拡張しても良い。全てのデータおよび制御構造は、初期化中の走査が一般的に成されないように特別に構成される。殆ど全ての制御構造（WSLおよびWBLを除く）は、通常、CB内に記憶された対応する情報から得られる。カードの初期化中においては、以下の操作を実行することが必要である。

1. ブートブロックから最後の制御ブロックポインタテーブルエントリーを読み込み、CBブロックの場所を確認する。
2. CBを操作することによってTB L Pを再構成する。
3. 制御ブロック内の最後のTB L エントリー内で規定されるライトポインタの位置の後に連続的に続くページのヘッダー／E C C領域を走査して、最後のS A T書き換え後に書き込まれたセクタを認識するとともに、WS LおよびW B Lを再構成する。
4. CB内の対応するM A PエントリーからN E Bを構成する。
5. C O BおよびA S B Lを構成する。
6. 最後のM A PのErB領域内で参照されたブロックが実際に消去されたか否かをチェックする。消去されていない場合には、消去操作を終了する。

【0134】

WS LおよびW B Lの構成

カードの初期化中、記憶されるライトポインタ（WP）の最後の値は、CB内の最後のTB L エントリーから読み込まれ、その場所の次のページで、ページヘッダーの走査が実行されて、プロセッサのS R A M内でWS LおよびW B Lが再

構成される。消去場所が見つかり、最後のSAT書き換え後に書き込まれたセクタ系列の末端に辿りつく。

このようなセクタ走査では、ライトポインタ(WP)が、ブロックの末端から、隣接していないブロックの先頭へとジャンプすることができる、という事実を考慮に入れなければならない。WPによって成される全てのブロック移動は、CB内のMAPエントリーのリンク領域内に記録される。

【0135】

COBおよびASBLの構成

これらの構造は、対応するエントリーをCBからコピーすることによって再構成することができる。また、COB構造を構成するためには(その中の削除セクタを認識するためには)、CB内の最後のMAPエントリーの廃棄(Obs)領域内にアドレスが規定されているこれらのセクタを含む現在のブロックを走査する必要がある。このブロック内の廃棄セクタを確認するためには、WSLおよびWBLを走査することも必要である。ASBLページを記録するためには、TBLからASBアドレスを認識し、その後、それらのヘッダー/ECC領域を走査しなければならない。

【0136】

電力損失リカバリ

メモリスシステムにおいては、正常に動作でき、また、電力が回復された際、電力が除去された状況下であっても、記憶されたデータが失われないことが必要である。しかしながら、電源ONの直後にメモリスシステムが全ての正常状態を回復する必要はなく、正常に動作できるだけで良い。任意の異常状態が何時検知されようとも、正常状態は、例外として、その後に回復され得る。

【0137】

以下の任意の操作が実行されている時に供給電圧が除去されると、メモリスシステムの正常状態が低下する。

1. ホストからのデータセクタの書き込み
2. 移動されるデータセクタの書き込み
3. 制御データブロック(CBまたはBB)へのエントリーの書き込み

4. 制御データブロック (SATまたはCB) へのページの書き込み
5. 廃棄セクタもしくは制御データを有する任意のブロックの消去

【0138】

ホストからデータセクタを書き込んでいる間の電力損失

この場合、書き込まれているデータが失われることがあるが、書き込みコマンドが終了してセクタを再び書き込むことができるといった情報は、ホストに与えられない。不完全に書き込まれたセクタは、不完全書き込み操作の結果として、フラッシュメモリ内に存在している場合がある。これは、CB内の最後のリンクパラメータによって規定されるブロック内のページヘッダーを読み込むことによってライトポイントの値が設定されると、初期化中に検知される。最後に検知されたセクタは、そのECCをチェックするために完全に読み込まなければならない。また、次のページは、それが完全に消去されることをチェックするように読み込まなければならない。不完全に書き込まれたセクタが検知される場合には、他の全てのセクタは、次の消去ブロックの最初に、新たなライトポイント位置へと再配置され、その後、ブロックが消去されなければならない。

【0139】

再配置されるデータセクタの書き込み中の電力損失

これは、前述したように、初期化中、ライトポイントを定めるプロセスにおいて検知される。セクタを再配置させてブロックを消去する場合と同じ動作が成されなければならない。また、ライトポイントの直前の論理セクタアドレスと、CB内の廃棄 (Obs) パラメータによって規定されるブロック内の廃棄セクタの論理セクタアドレスとを比較することによって、不完全な再配置操作が検知されなければならない。保留の任意のセクタ再配置は、初期化中に完了されなければならない。

【0140】

制御データブロック (CBまたはBB) へのエントリーの書き込み中の電力損失

この状態は、CBおよびBB内のエントリーが読み込まれ且つそれらのECCがチェックされると、通常の初期化中に検知される場合がある。エントリーのECCがエラーを示している場合には、そのエントリーを無視すべきである。CB

またはBBエントリー書き込み操作を開始した初期の動作は正確に完了されず、この操作が通常の動作中に繰り返される時、エントリーは、その後、正確に書き込まれる。

【0141】

制御データブロック(ASB)へのページの書き込み中の電力損失

この状態は、ASB内のページが読み込まれ且つそれらのECCがチェックされると、通常の初期化中に検知される場合がある。ページのECCがエラーを示している場合には、そのページを無視すべきである。ASBページ書き込み操作を開始した初期の動作は正確に完了されず、この操作が通常の動作中に繰り返される時、ページは、その後、正確に書き込まれる。

【0142】

全ての制御データブロック(SATまたはCB)の書き込み中の電力損失

これによって、不完全な制御データブロックが、他のデータ構造によるそれとは無関係に、フラッシュメモリ内に存在するようになる。この状態が初期化中に検知される必要はない。また、ブロックは、「損失ブロック」として存在することが許されても良い。ブロック書き込み操作を開始した初期の動作は正確に完了されず、この操作が通常の動作中に繰り返される時、ブロックは、その後、正確に書き込まれる。通常の動作の後段で、損失ブロックは、そのMAP状態に関する矛盾、あるいは、システム内の消去ブロック数における矛盾を発見することによって検知される(図16のキャパシティマップ参照)。その後、必要に応じて、除外ルーチンは、フラッシュメモリ全体を走査することによって、そのブロックを認識して消去する。

【0143】

廃棄セクタまたは制御データとともにブロックを消去する間の電力損失

これによって、不完全に消去されたブロックがフラッシュメモリ内に存在するようになる。この状態は、CB内の最後のMAPエントリー内のErB領域によって参照されるブロック状態がチェックされると、初期化中に検知される。必要に応じて、このブロックの再消去を実行することができる。

【0144】

更なる別の実施例

本発明の範囲から逸脱することなく、前述した実施例を様々に変形することができる。例えば、消去操作を扱う他の方法は、2つのCOB（および、2つのCOSB）の存在を常に許容することである。この利点は、メモリ容量を最大限に利用できることである。前述した実施例においては、1つのCOBだけが許容されたが、COBを作りたいブロック内にライトポインタがある場合には、1つのPOBの一時的な存在を許容しても良い。これは、POBを有することが必要な場合に、POBのために割当てることができる十分な消去メモリ容量が常に存在していなければならないことを意味する。したがって、このメモリ容量を最大限に利用できることは有益であり、これを実現する1つの方法が、2つのCOBの存在を常に許容すること、すなわち、POBを不要にすることである（必要な場合、第2のCOBがPOBとして作用することができる）。そのような2COBシステムにおいて、新たなCOBを形成することが必要な場合には、2つのCOBのうちの古い方のCOBを消去する（古い方のCOB内にライトポインタが無い場合には、新しい方のCOBを消去する）。

【0145】

図3、および、メモリシステム内の各フラッシュページにおけるデータ配置の説明に関し、以下、ページ内にデータを記憶する幾つかの別の方法を提案する。図28は、典型的な528バイトのNAND型もしくはAND型フラッシュメモリのページ1の物理的な群分離を示している。ページは、512バイトの「データ領域」300と、16バイトの「スペア領域」302とを備えている。図3に関して前述した実施例において、コントローラ8は、データ領域300内に512バイトの情報1a（例えば、1つのホストデータセクタ）を記憶し、スペア領域302内にヘッダー1bおよびECC1c（以下、これらをまとめて、オーバーヘッドデータと称する）を記憶する。しかしながら、ページ1内における他のデータ配置も可能である。例えば、図29に示されるように、データ領域300の第1の部分303内にヘッダ1bおよびECC1cが記憶され、データ領域300の残りの部分とスペア領域302とから成る部分304内に情報1aが記憶されても良い。

【0146】

図30に示されるように、フラッシュページの先頭からオフセットした位置にヘッダー1bおよびECC1cを書き込むとともに、残っているヘッダーおよびECCの両側の空間内にホストデータセクタ（「ユーザデータ」と称しても良い）を書き込むこともできる。ODのオフセット量（オフセットS）は、例えば、

（a）ページ1の物理アドレス（PA）、あるいは、（b）ページ1に書き込まれるユーザデータ（すなわち、ホストデータセクタ）の第1のバイト内の1または複数のビットに応じて決定されても良い。図31（a）は、セクタ書き込み操作の開始前におけるコントローラバッファ320内のデータの配置を示している。データは、ユーザデータの第1の部分332として、また、ヘッダデータの第2の部分324として配置されている。図31（b）は、オフセットSがユーザデータの最初のバイト内の1または複数のビットによって決定される（前述した選択（b））書き込み操作の終了後における、フラッシュメモリページ内のデータの配置を示している。データは、ユーザデータから成る第1の部分326として記憶され、その後、ユーザデータから成る第2の部分328として記憶され、その後、ヘッダー1bおよびECC1cとして記憶され、その後、ユーザデータから成る第3の最終部分330として記憶される。部分326と部分328とを足した長さは、部分326内のデータに依存する。部分326の長さは最小オフセット以下に規定され、部分328の長さは、正確なオフセットSを与えるために、部分326内のデータに基づいて演算される。ユーザデータから成る第1および第2の部分326、328は別個に認識され、これにより、第1の部分326は、1つの操作でコントローラによりフラッシュメモリから読み込まれるとともに、次の操作でコントローラによって読み込まれる第2の部分328の長さを決定するためにコントローラによって演算される。図32は、書き込み操作中にコントロールバッファからフラッシュメモリにデータを送信するために使用されるコントローラコマンドのシーケンスを詳細に示した表である。

【0147】

ユーザデータの1または複数のビットの関数であるオフセットSを選択する1つの利点は、全てのセクタにおいて、オーバーヘッドデータが528バイトデー

タセグメント内の同じ位置に挿入されないという点である。これにより、メモリアレーのカラム欠陥といったフラッシュメモリ内のシステム故障時に、多数のセクタにおいて、貴重なオーバーヘッドデータが同時に失われることが防止される。

【0148】

図33は、図31(b)のフラッシュメモリページにおける読み込み操作終了後のコントローラバッファのデータ配置結果を示している。バッファ内のデータが配置された後に、ユーザデータから成る第1の部分322およびヘッダーデータから成る第2の部分324が配置され、その後、ECCから成る第3の最後の部分325が配置されているのが図33から分かる。図34は、読み込み操作中にフラッシュメモリからコントローラバッファへデータを送信するために使用されるコントローラコマンドのシーケンスを詳細に示す表である。

【0149】

さらに、複数のフラッシュチップに対するインターリーブ書き込み操作に関しては、マルチプルフラッシュチップメモリシステムにおいて既に説明したように、複数のチップに対して略同時に書き込む技術は、物理的なページサイズがコントローラによるセクタ書き込みの多様なサイズである単一のメモリチップにデータを書き込むために使用することもできる。例えば、コントローラが均一のサイズのセグメントにデータを書き込む場合、メモリの各ページは、コントローラによって書き込まれる（ユーザ+オーバーヘッド）データのセグメントサイズの4倍である。

【0150】

本発明は、NAND型メモリだけでなく、AND型およびNOR型といった他のタイプのメモリにも適用できる。AND型フラッシュメモリの場合、ブロックの各ページは、図28のNANDページフォーマットと同じフォーマットを有し、前述したようにページ内でデータを任意に配置することができる。本発明のコントローラは、たとえ、不良セクタを含むブロック内で、消去されるそのブロック内の個々の良好なセクタが個別に消去される場合であっても、セクタのブロック内のメモリを依然として消去する。すなわち、コントローラは、不良セクタを

含む任意のブロックを不良セクタとして処理せず、その代わり、それらを良好な（消去可能な）ブロックとして処理し、これらのブロック内の良好なセクタを利用する。しかしながら、AND型の実施例において、コントローラは、SATブロックもしくはASBsに関し、全て良好なセクタを含むブロックのみを使用する。

【0151】

AND型フラッシュメモリが使用され且つメモリシステムが前述したようにインターリーブチップ書き込み操作を使用するマルチプルフラッシュチップシステムである場合、また、仮想ブロックのうちの1つの任意のセクタブロック（ページ）が不良セクタを含んでいる場合、コントローラによって、ライトポインタは、このセクタをスキップして、ブロック内の次の良好なセクタへと進む。例えば、cをチップとし、sをセクタとした場合、4つのセクタ書き込みのバーストがc3s5、c4s5、c1s6、c2s6であって、c1s6が不良セクタであると、順序は、c3s5、c4s5、c2s6、c3s6となる。これは、1つの仮想ブロック内の1つのブロックが1または複数の不良セクタを含んでいる場合にコントローラがそのブロックを不良ブロックとして処理し且つ仮想ブロック全体を不良仮想ブロックとして処理するNAND型メモリに基づく実施例と対称的である。

【0152】

NOR型フラッシュメモリを使用する場合、好ましい実施例において、メモリシステムのコントローラは、均一サイズの528バイトセクタの状態、データ構造をフラッシュメモリに書き込んだり、データ構造をフラッシュメモリから読み込んだりする。図35は、そのような3つのセクタ1、2、3がNORメモリのブロック4'内にある状態を概略的に示している。NORブロック内の1列のメモリがたった512バイトであるため、NOR内の各セクタは、1列を満たすとともに、次の列を満たすように回り込むようになる。しかしながら、NORメモリ内のセクタを異なった方法で規定することもできる。528バイトを下回る或は上回るサイズのセクタを使用することもでき、また、ブロックは、ワンサイズ以上大きいセクタを含んでいても良い。コントローラは、NAND型およびA

NAND型メモリページに関して既に述べた任意の様々な異なる方法により、ユーザおよびオーバーヘッドデータを含む各セクタ内でデータを配置しても良い。

【0153】

前述したことから分かるように、メモリシステムがNAND型、AND型、NOR型のメモリアレーである場合、メモリシステムの物理セクタは、メモリアレーそれ自身の物理アーキテクチャと特定の関係を有している必要は全くない。例えば、セクタがアレーの列（ページ）に対応している必要はなく、また、1つのセクタとこれに隣接するセクタとの間の境界を認識できるように物理的特徴が存在している必要もない。常に1つのユニットとしてコントローラにより処理されるメモリセル群として、セクタを解釈することができる。異なるセクタは同じサイズを有している必要はない。また、セクタの物理的構造は、セクタ内に記憶されるデータに依存しない。また、メモリセルの列（ページ）内の欠陥領域が容認され且つ物理セクタへの書き込み時にコントローラによって飛び越えられるような実施例も可能である。

【0154】

SATに関しては、前述したように、SATが好ましくはメモリアレー6の1または複数のブロック内に記憶される一方で、コントローラにアクセス可能な別個の不揮発性メモリをメモリシステム10に設けることもできる。この場合、コントローラは、この別個のメモリ内にSATを記憶する。

【0155】

最後に、前述した実施例の変形バージョンにおいては、前述したように、利用可能な消去ブロックを常にその昇順で使用する代わりに、コントローラは、消去ブロックを他の順番で使用する。この変形例において、NEBリストは、現在利用可能な全ての消去ブロックの選択されたサブセットを含んでおり、NEBリスト内の最初のブロックアドレスが、使用される次の消去ブロックである。この最初のブロックアドレスは、それがデータ記憶の用途で割当てられた時に、NEBリストから除去される。形成される任意の新たな消去ブロック（例えば、ホストからの削除コマンドに続く、旧データの形成）は、NEBリストの低部に加えられる。これは、コントローラ（例えば、ホストからの所定数のセクタ書き込みコ

マンドであっても良い) によって決定される所定の期間だけ続く。そして、その期間の終端で、コントローラは、NEB内のエントリーを現在利用可能な消去ブロックの新たなサブセットに置き換えることにより、NEBリストを再びコンパイルする。全ての消去ブロックの全セットから成るサブセットは、物理ブロックアドレスを上昇する順序で都合良く順次に使用されても良い。この変形例は、全てのブロックの消去状態をモニタして記憶することに伴うメモリ空間の要件を低減させる点において、幾つかの利点を有する。

【図面の簡単な説明】

【図1】

その中に3つのセクタを示すNAND型フラッシュメモリのセクタの1つのブロックの概略図である。

【図2】

フラッシュチップとコントローラチップとを備えるメモリシステムのブロック図である。

【図3】

NAND型もしくはAND型のフラッシュメモリ内のデータの1ページの概略図である。

【図4】

図3のページのヘッダー領域の構造を示す図である。

【図5】

1ページの物理アドレス(PA)のフォーマットを示す図である。

【図6】

制御ブロック(CB)エントリーを示す図である。

【図7】

テーブルブロックリスト(TBL)内の1つのエントリーを示す図である。

【図8】

MAPエントリーのフォーマットを示す図である。

【図9】

ASBリスト(ASBL)内の1つのエントリーのフォーマットを示す図であ

る。

【図10】

カレントオブソリートブロック (COB) 構造のフォーマットを示す図である。

【図11】

本発明の一実施例に係るマルチプルフラッシュチップメモリシステムの仮想ブロック内にセクタが書き込まれる順番を示すテーブルである。

【図12】

仮想アドレス (VA) のフォーマットを示す図である。

【図13】

VAからのPAの取得方法を示す図である。

【図14】

本発明に係るマルチプルフラッシュチップメモリシステムに複数のセクタを書き込む間の操作のタイミングを示す図である。

【図15】

コントローラチップのブロック図である。

【図16】

本発明のメモリシステムのために割当てられたメモリ容量を示すテーブルである。

【図17】

アドレス変換プロセスを示すフローチャートである。

【図18】

図17のボックス58で実行されるステップのフローチャートである。

【図19】

4つのフラッシュチップおよびコントローラチップを備えるマルチプルフラッシュチップメモリシステムのブロック図である。

【図20】

図17のボックス56で実行されるステップのフローチャートである。

【図21】

図17のボックス44で実行されるステップのフローチャートである。

【図22】

セクタ読み込み操作のフローチャートである。

【図23】

セクタ書き込み操作のフローチャートである。

【図24】

図23のボックス161で実行されるステップのフローチャートである。

【図25】

図24のボックス207で実行されるステップのフローチャートである。

【図26】

図23のボックス160で実行されるステップのフローチャートである。

【図27】

セクタ削除操作のフローチャートである。

【図28】

NAND型もしくはAND型のフラッシュメモリ内のページの物理的な群分離を示す図である。

【図29】

図28のフラッシュページ内にデータを配置する他の方法を示す図である。

【図30】

図28のフラッシュページ内にデータを配置する更なる他の方法を示す図である。

【図31(a)】

セクタ書き込み操作前におけるコントローラのバッファメモリ内のデータを示す図である。

【図31(b)】

図30の実施例にしたがってデータが配置された場合における、書き込み操作終了後のフラッシュページ内のデータを示す図である。

【図32】

図31(a)および図31(b)の書き込み操作中に、コントローラバッファ

からフラッシュメモリにデータを送信するために使用されるコントローラコマンドのテーブルである。

【図33】

読み込み操作後におけるコントローラのバッファメモリ内のデータを示す図である。

【図34】

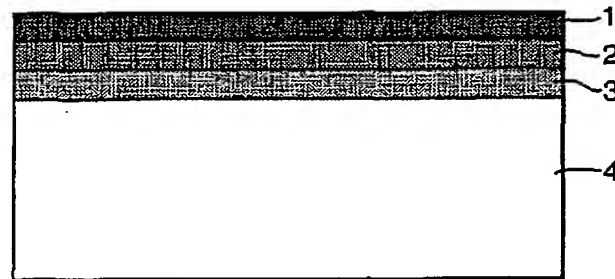
図33の読み込み操作中にフラッシュメモリからコントロールバッファにデータを送信するために使用されるコントローラコマンドのテーブルである。

【図35】

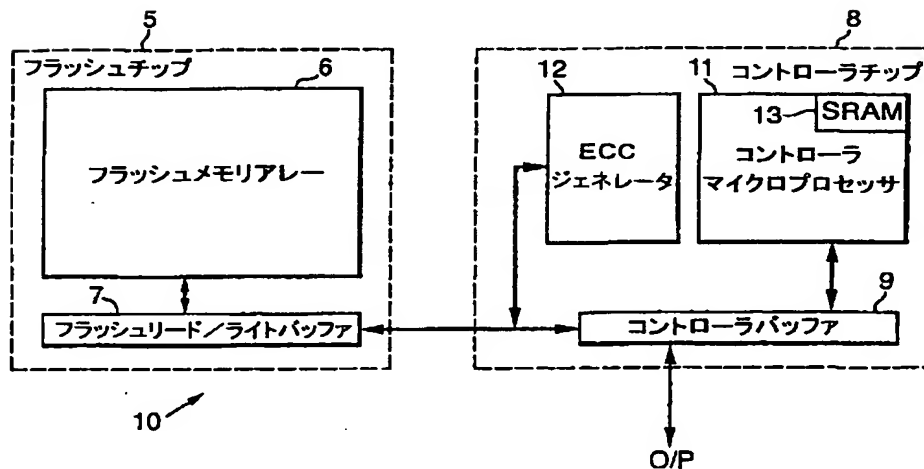
その中に3つのセクタを示すNOR型フラッシュメモリ内のセクタの消去可能ブロックの概略ブロック図である。

【図1】

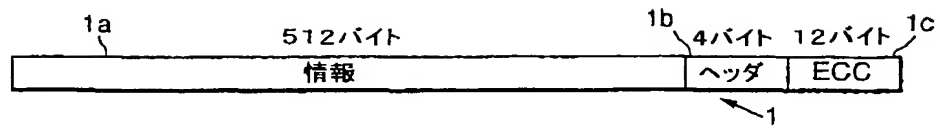
Fig.1.



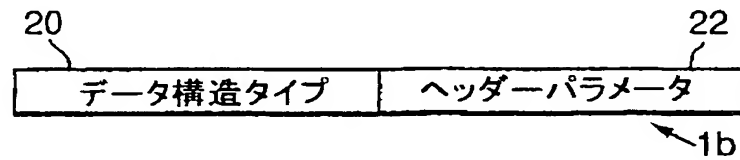
【図2】



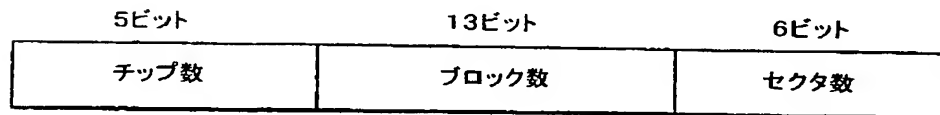
【図3】



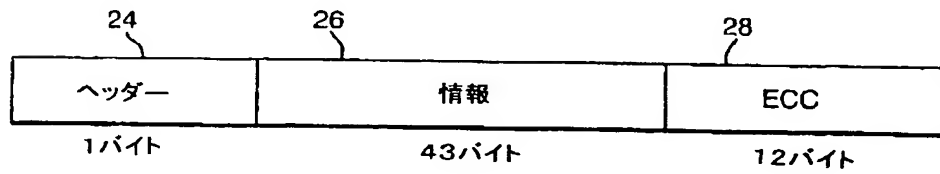
【図4】



【図5】



【図6】



【図 7】

No	RP	WP	SWP	保留	SAT _N	ASB _N	...	SAT _{N+7}	ASB _{N+7}
1バイト	1バイト	1バイト	1バイト	3バイト	18ビット	18ビット	...	18ビット	18ビット

【図8】

範囲	ObsC	ObsP	リンク	EB	ビットマップ
----	------	------	-----	----	--------

【図9】

SATブロック インデックス	SATブロック アドレス	ASB アドレス	LWP	NVP	ASBページ0	...	ASBページN
2バイト	3バイト	3バイト	1バイト	1バイト	1バイト	...	1バイト

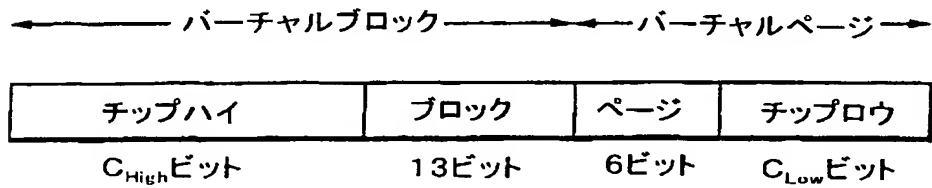
【図10】

28	ブロック数	30	廃棄もしくは削除セクタマスク
	4バイト		32バイト

【図11】

チップ0 ブロック0	チップ1 ブロック0	チップ2 ブロック0	チップ3 ブロック0
0	1	2	3
4	5	6	Etc.

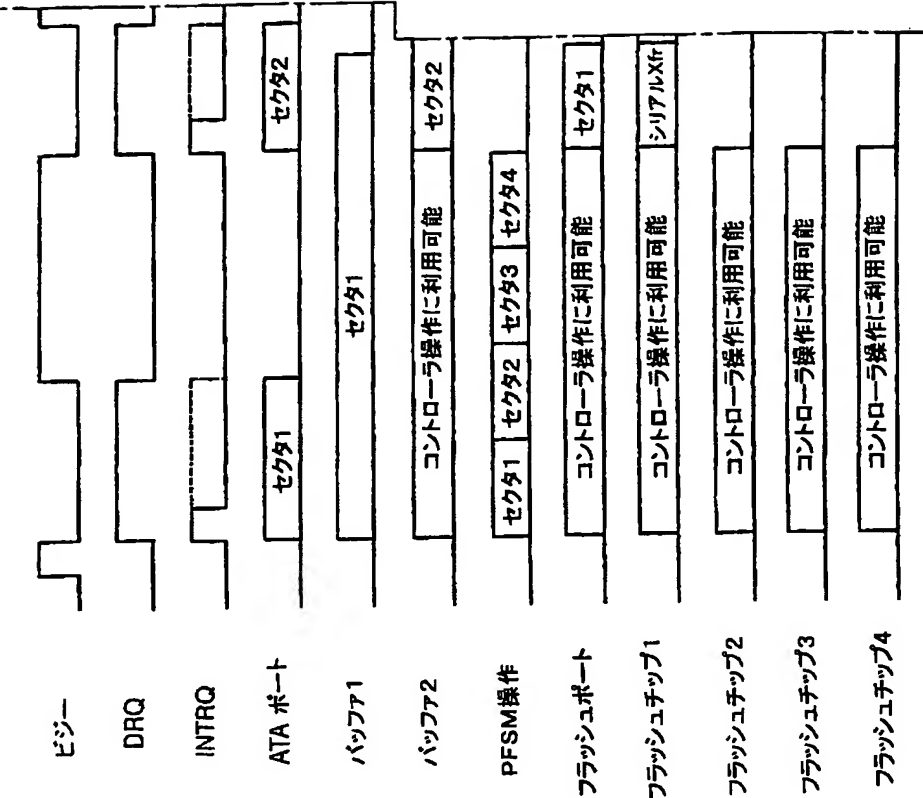
【図12】



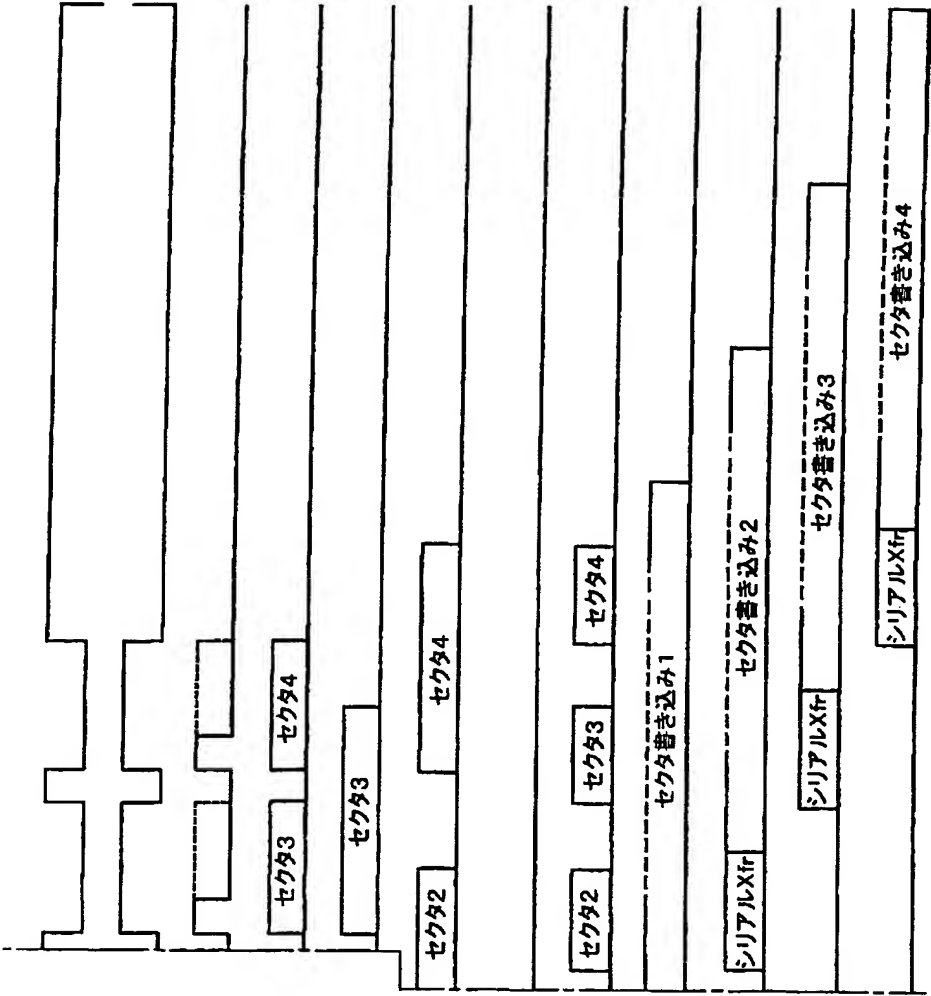
【図13】



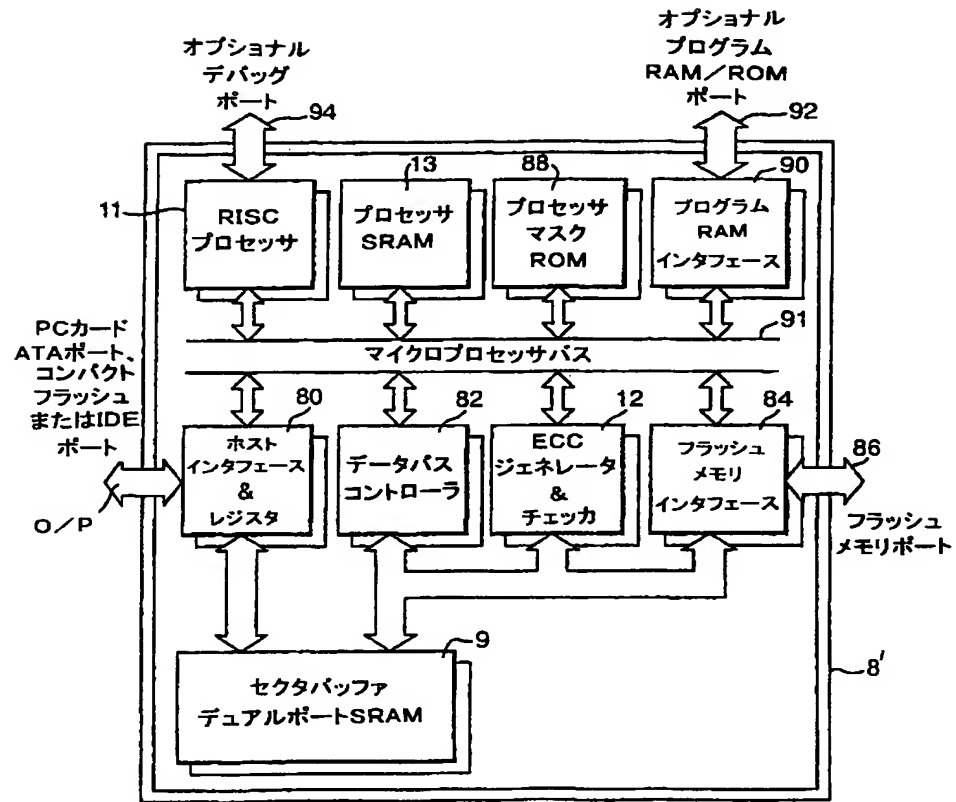
【図14】



【図14-1】



【図15】

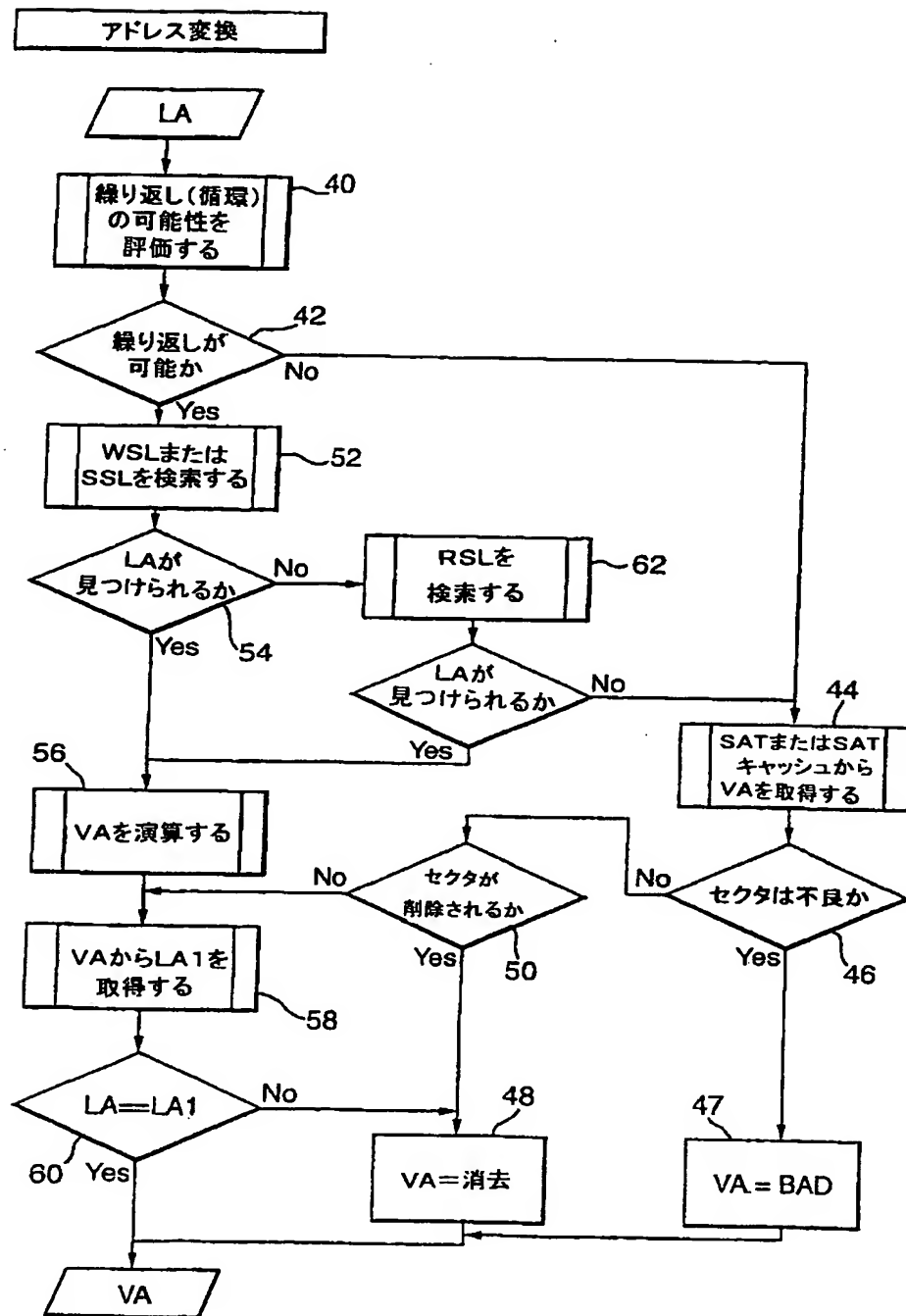


【図16】

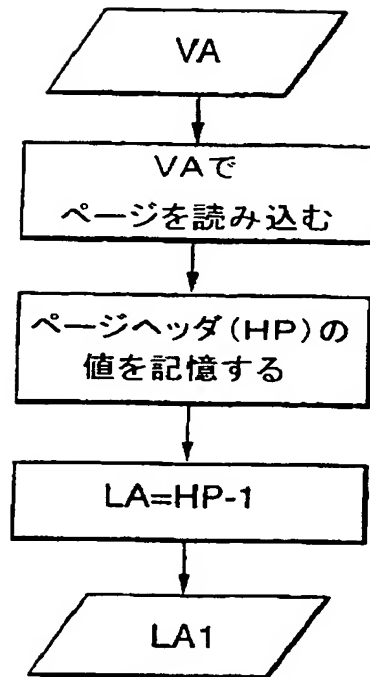
容量割り当てテーブル

	8MBカード		64MBカード		512MBカード	
	ブロックの 数	容量%	ブロックの 数	容量%	ブロックの 数	容量%
総容量	1024	100%	8192	100%	65536	100%
セクタ	1007	98.1%	8132	99.2%	65131	99.4%
ブートブロック	1	0.1%	1	0.01%	1	0.00%
制御ブロック	3	0.3%	4	0.05%	4	0.00%
セクタアドレステーブル	6	0.6%	48	0.6%	384	0.6%
追加SATブロック	6	0.6%	8	0.1%	8	0.01%
廃棄セクタ	1	0.1%	1	0.01%	1	0.00%
消去バッファ	2	0.2%	2	0.02%	2	0.00%
スペアブロック						

【図17】

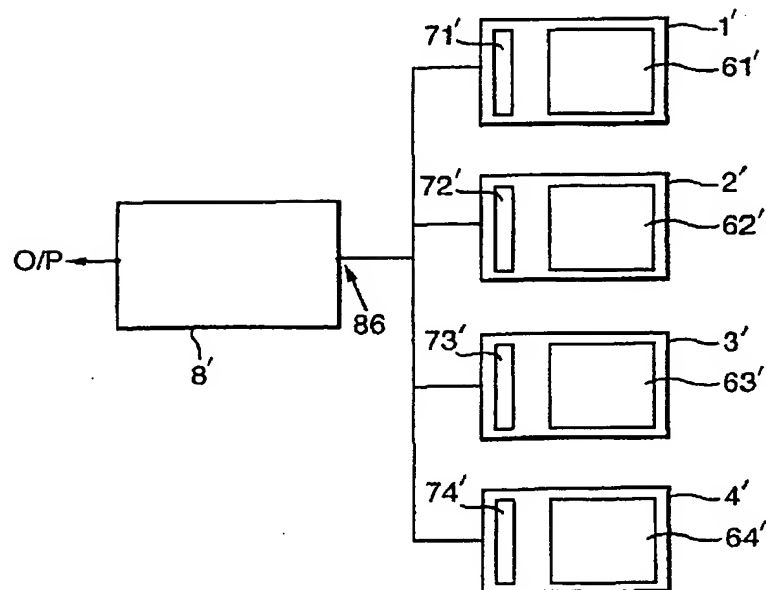


【図18】

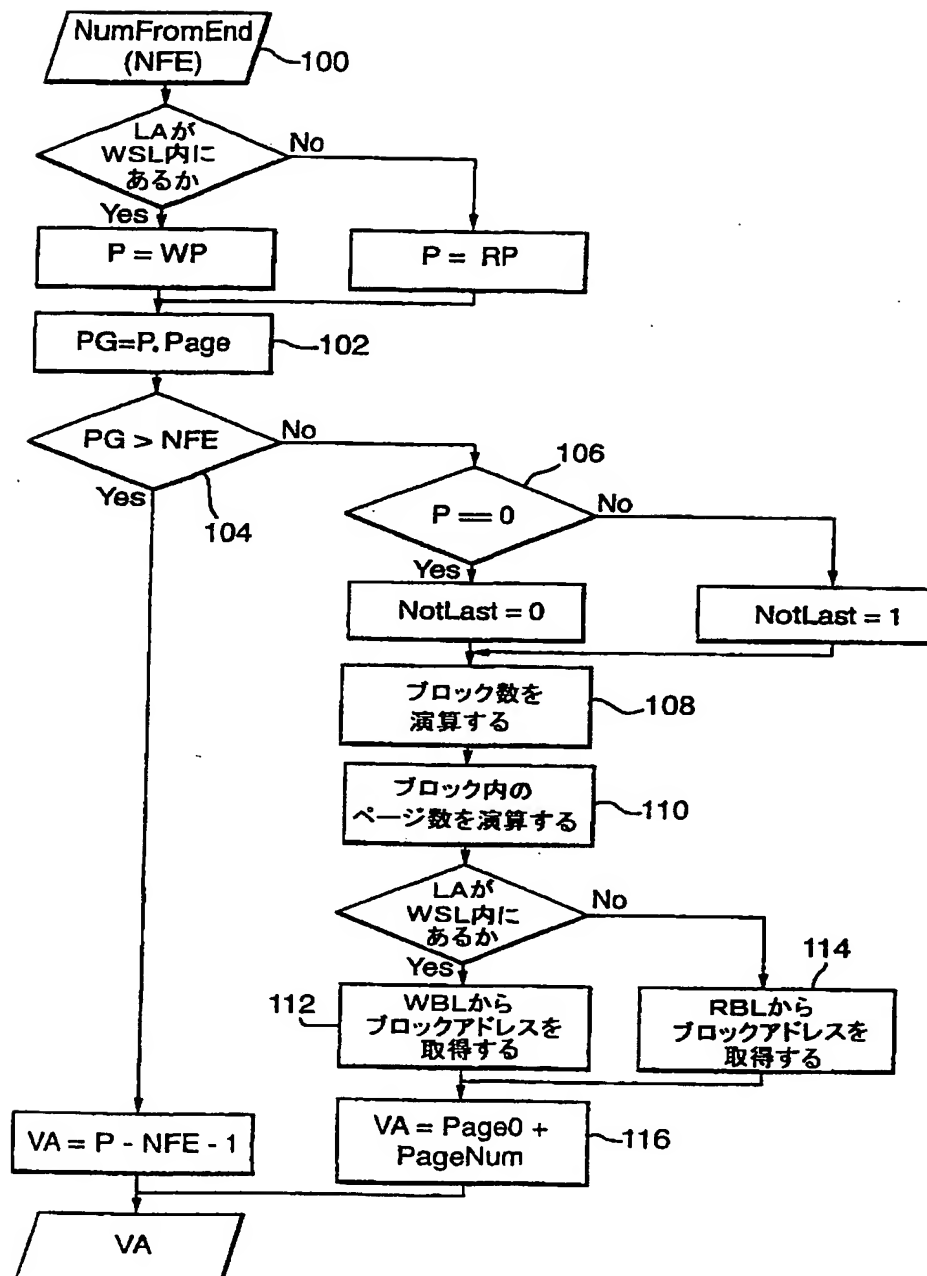


【図19】

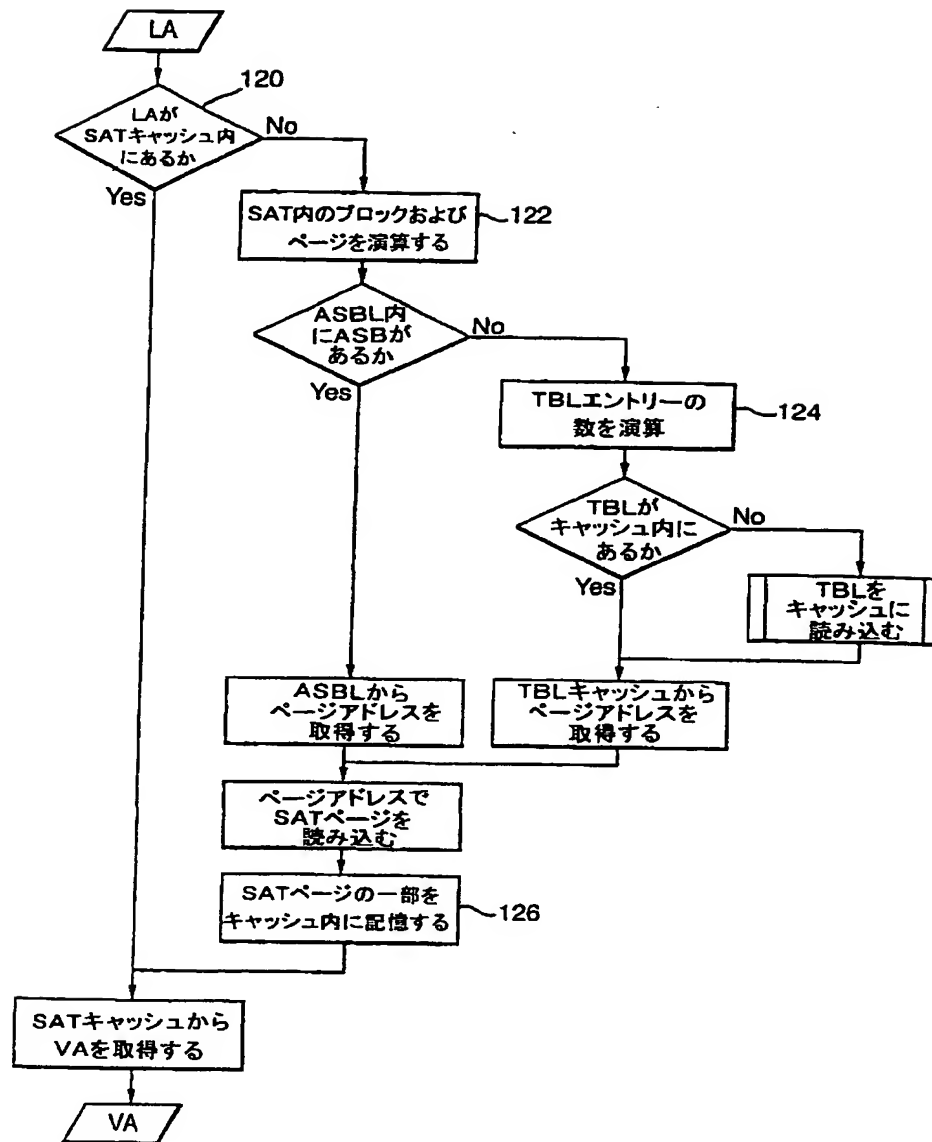
Fig.19.



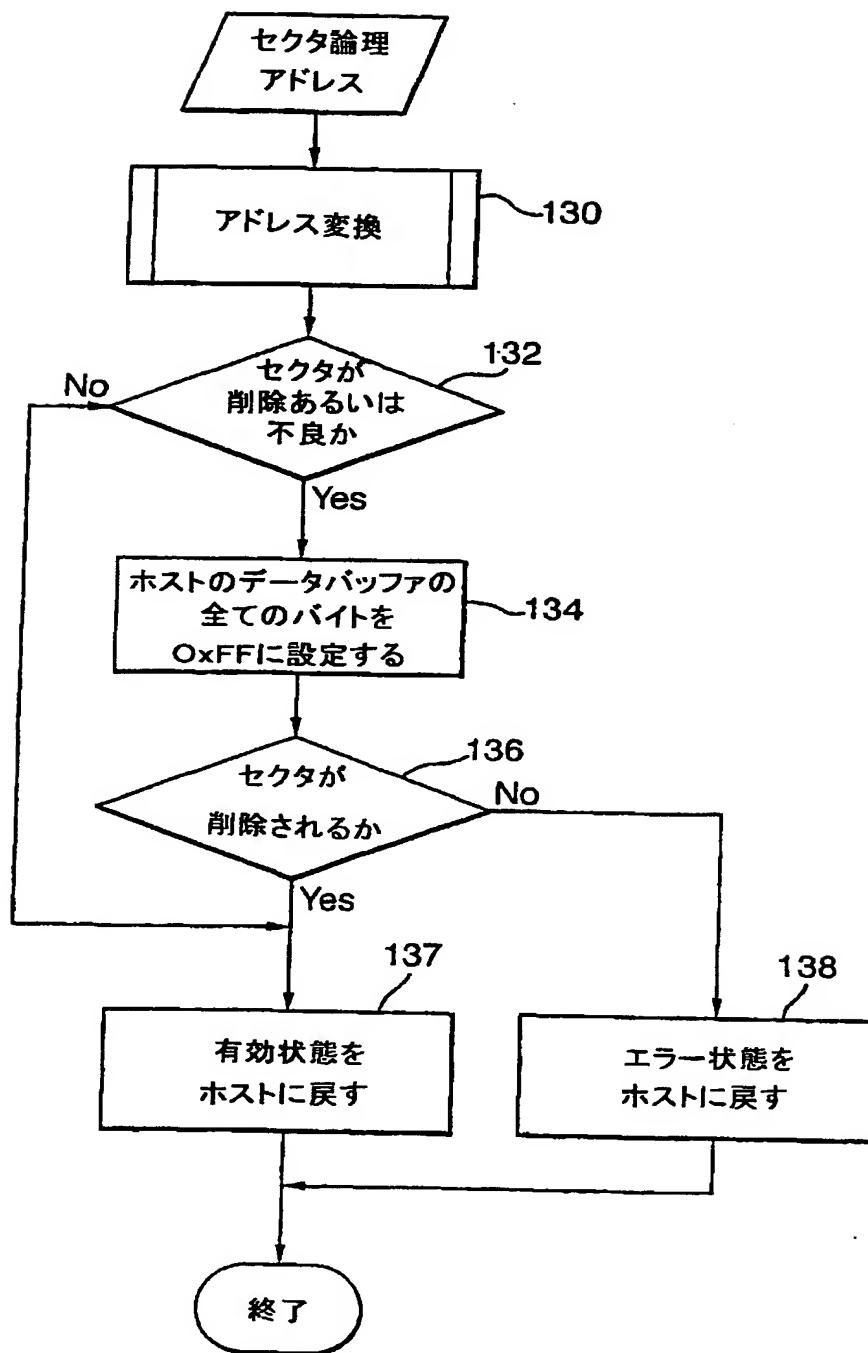
【図20】



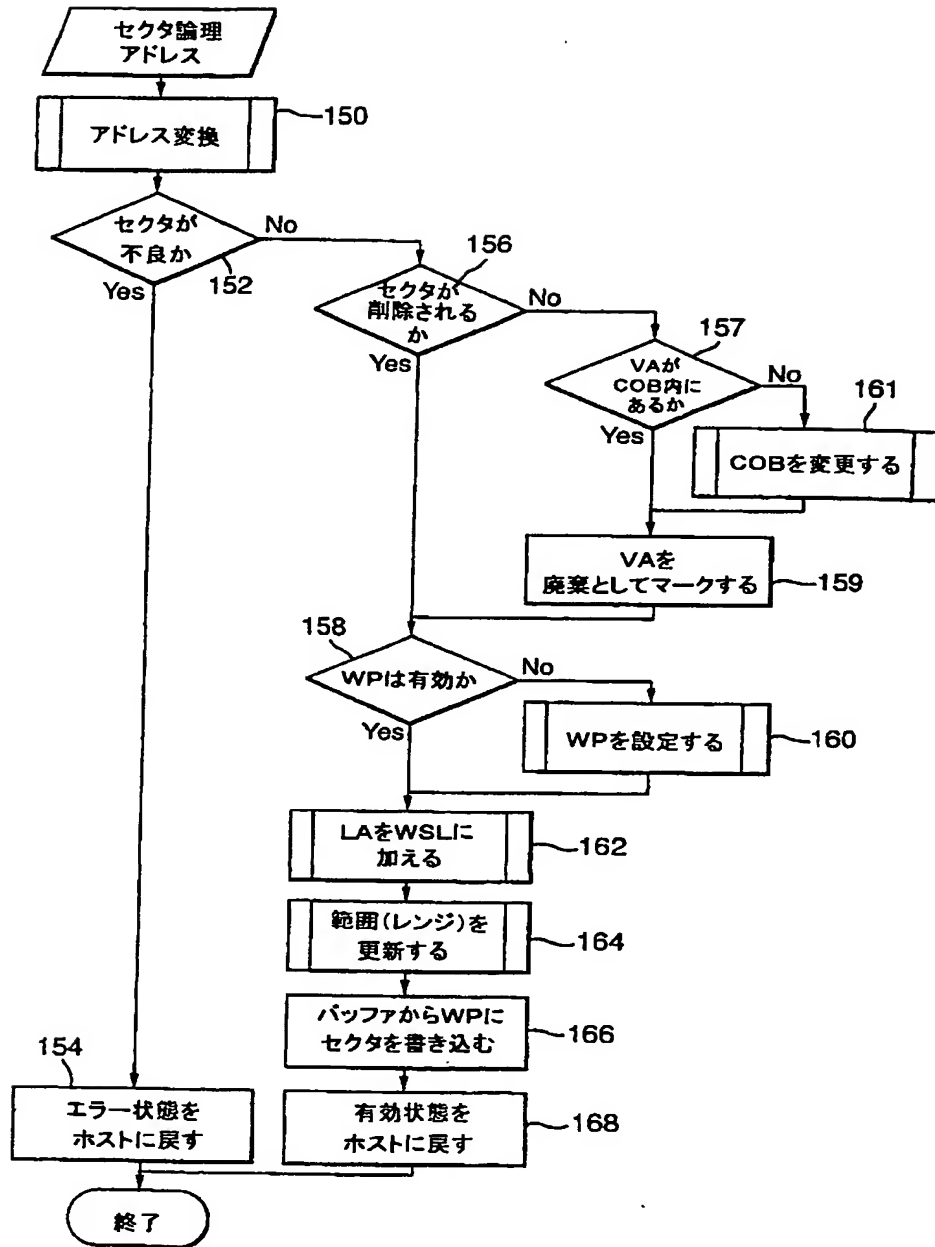
【図21】



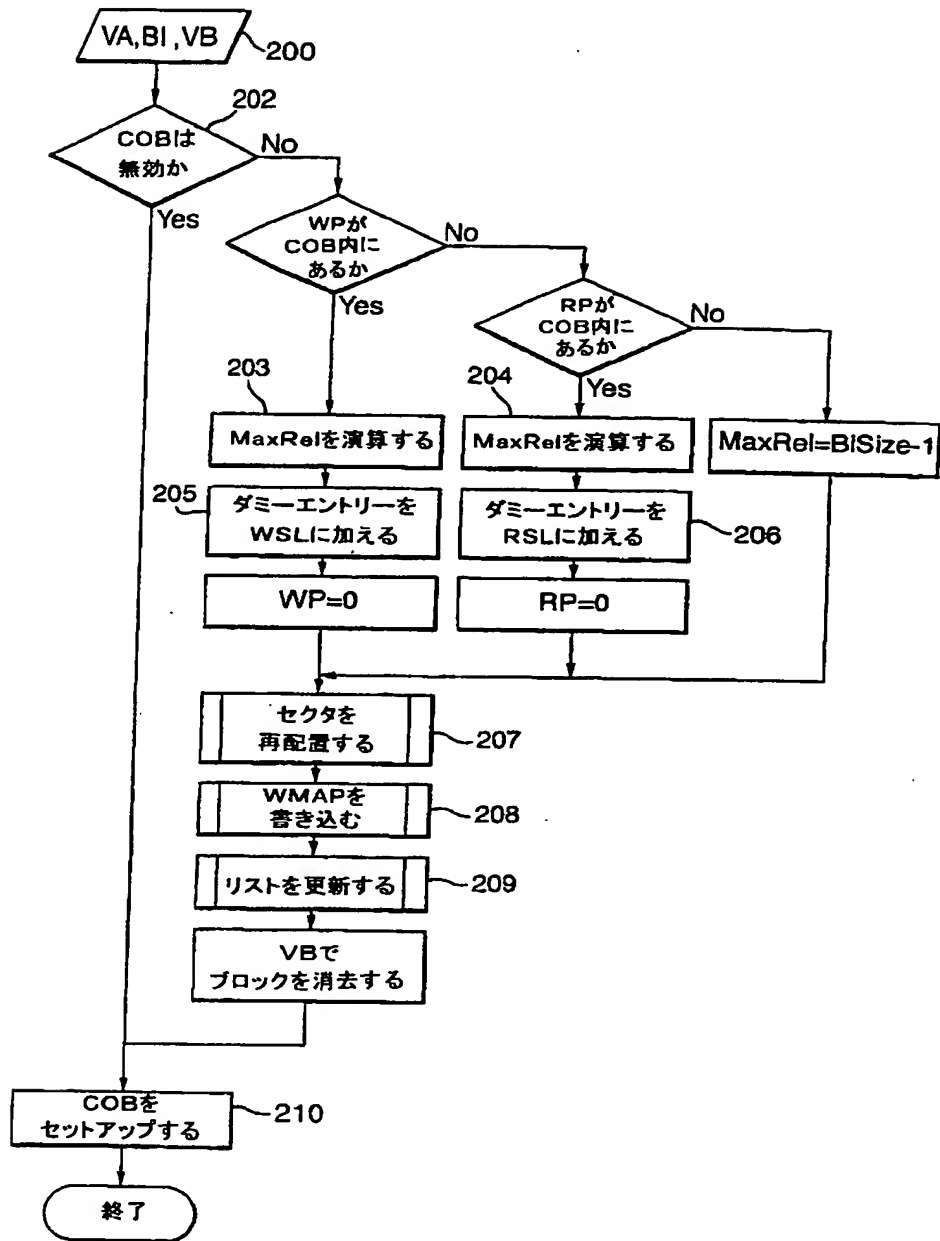
【図22】



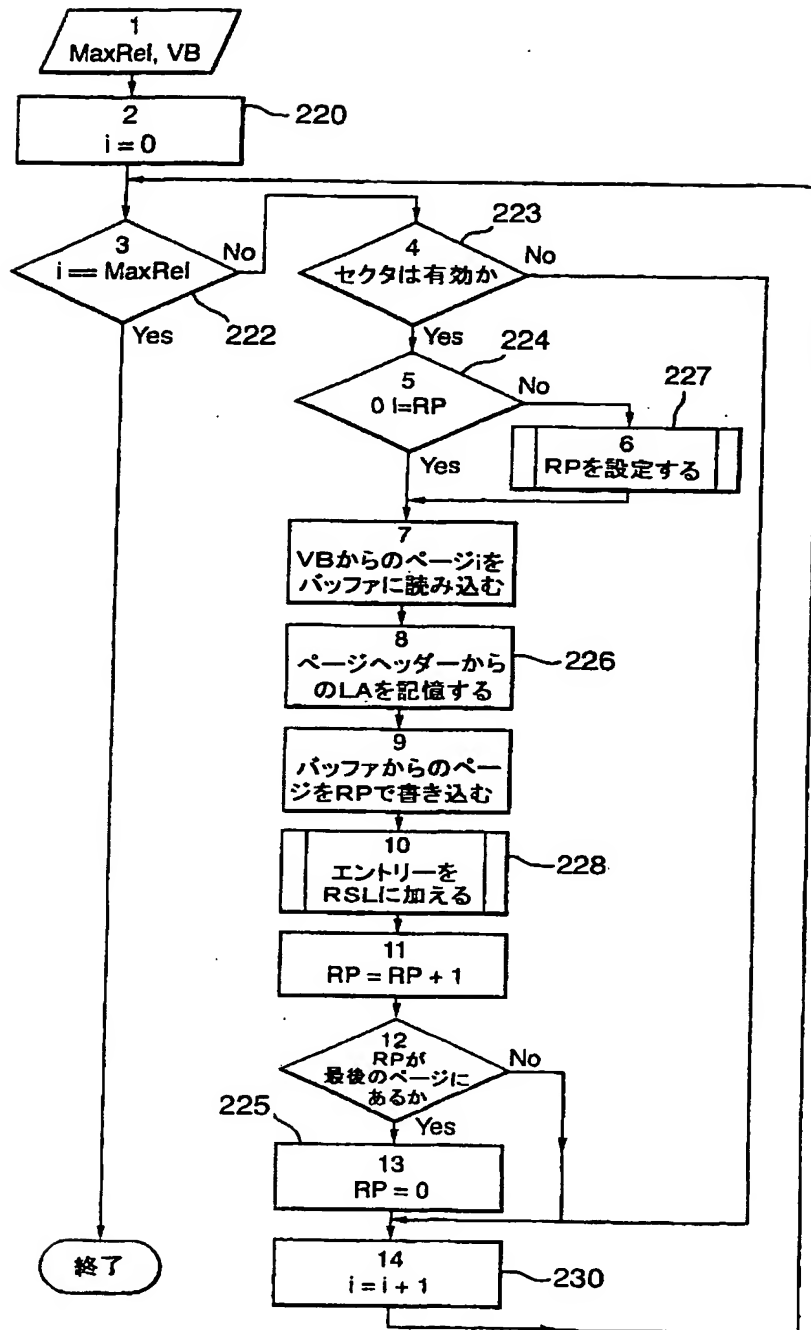
【図23】



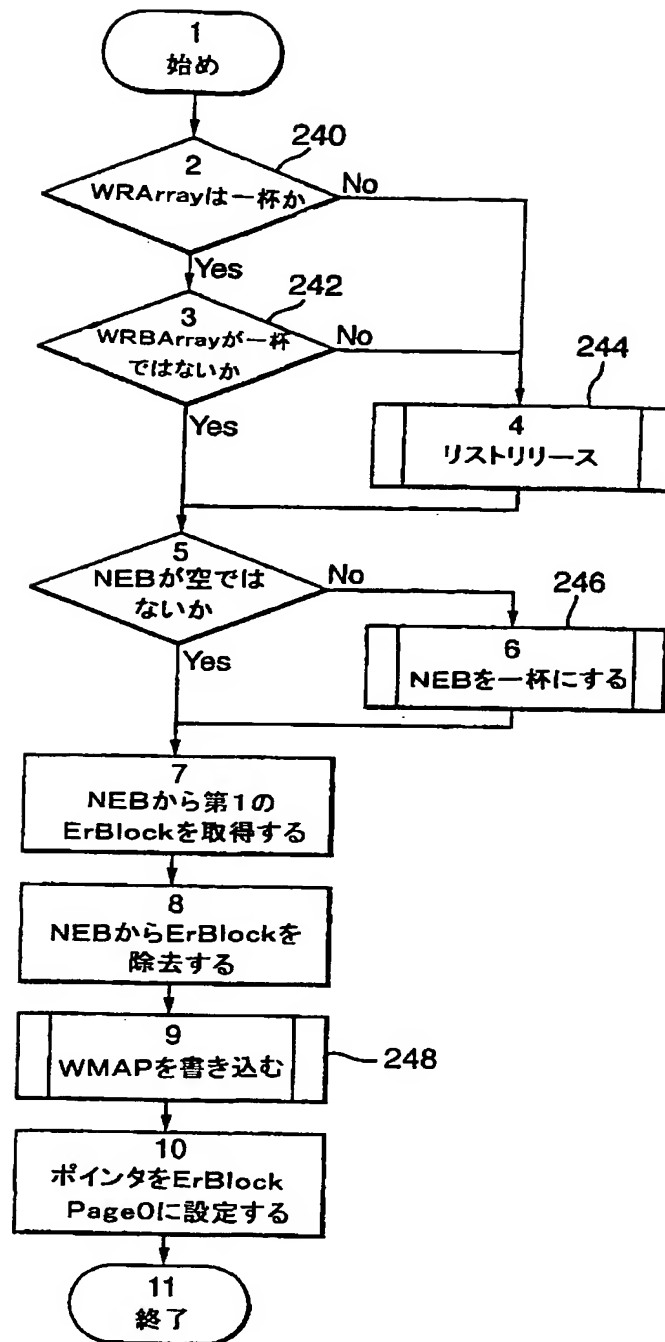
【図24】



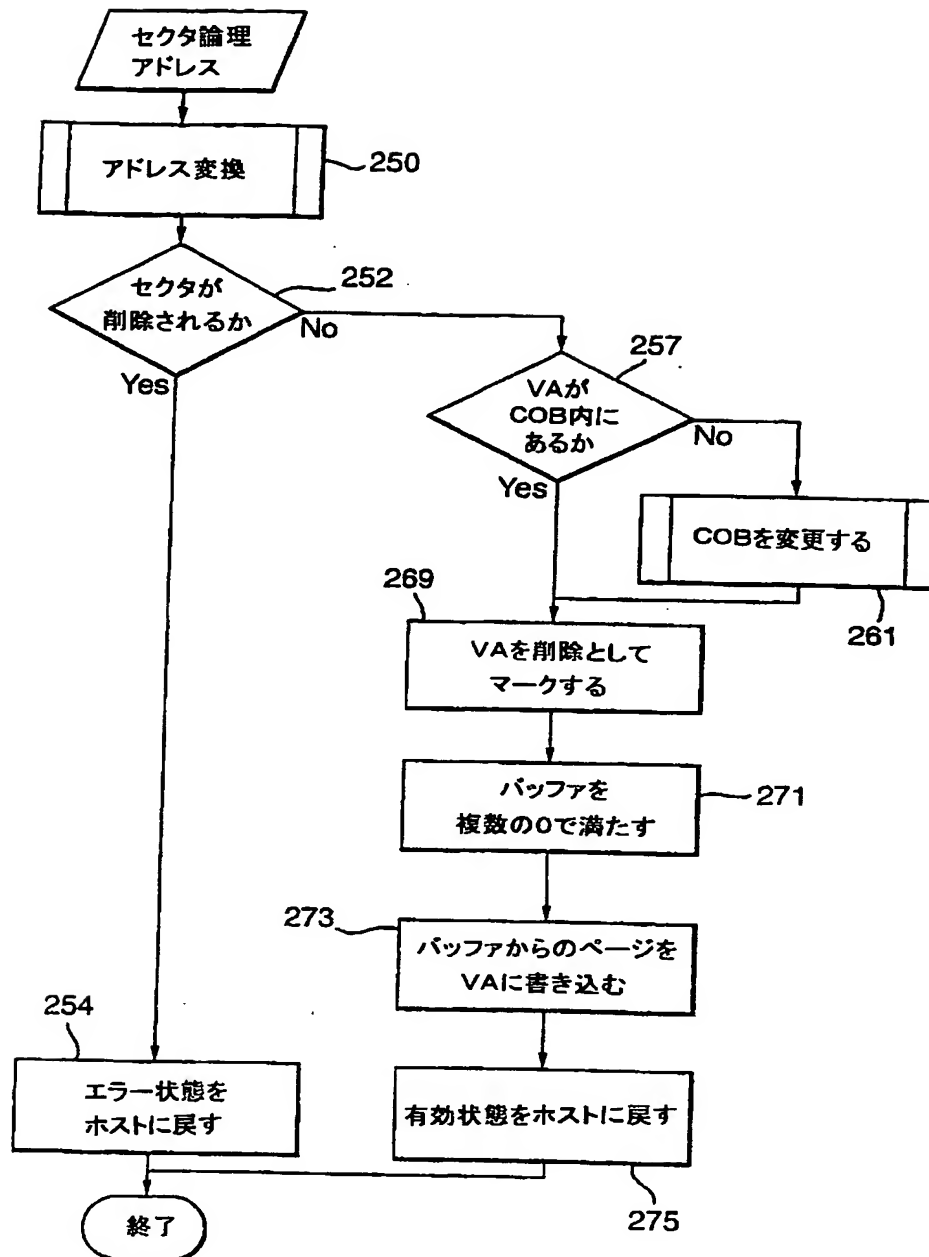
【図25】



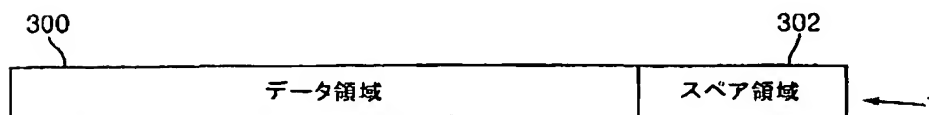
【図26】



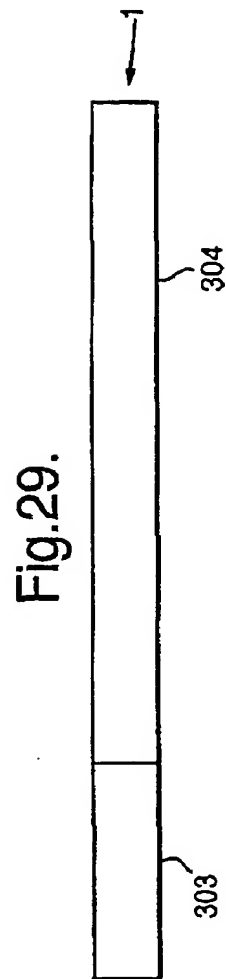
【図27】



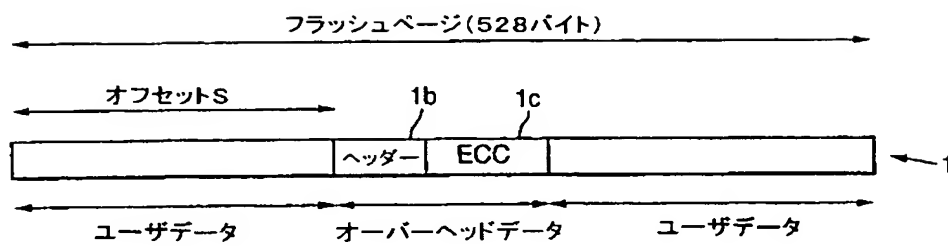
【図28】



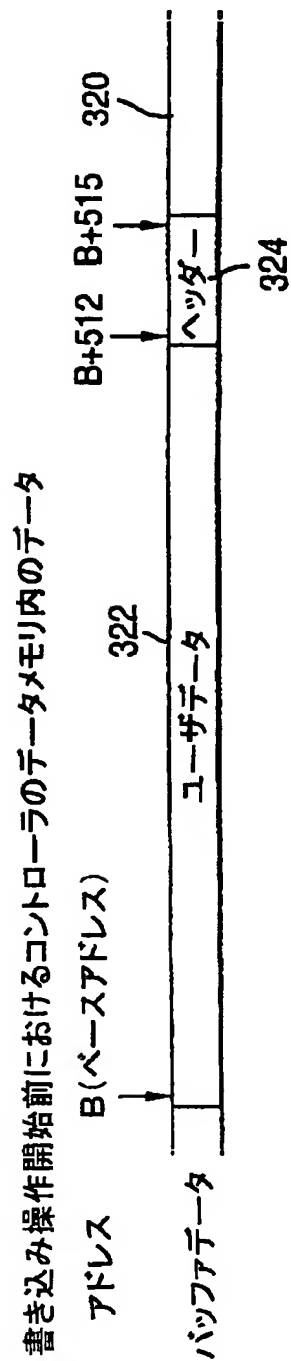
【図29】



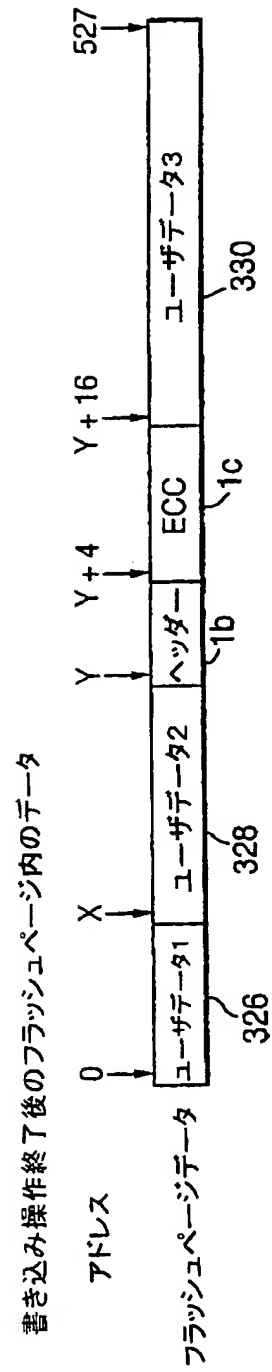
【図30】



【図31(a)】



【図31(b)】



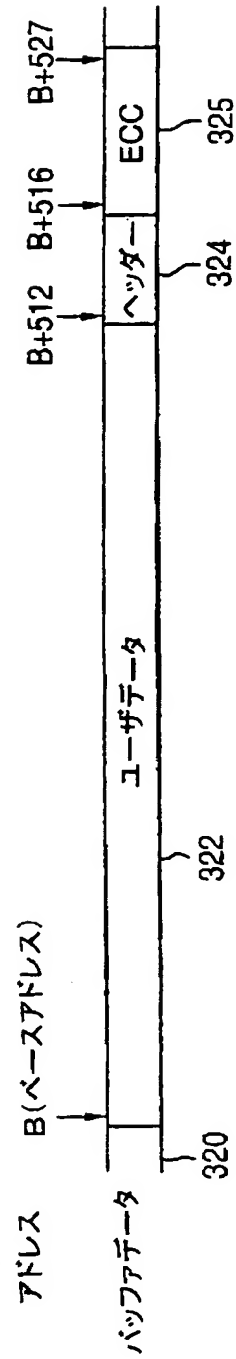
【図 3 2】

コントローラのデータ送信ハードウェアへのコマンドシーケンス

コマンド シーケンス	コマンドパラメータ			コマンドの内容		
	フラッシュ 開始 アドレス	バッファ 開始 アドレス	データXfr サイクル の数	ECCジェネレータモード	フラッシュ クロック	概要
Cmnd 1	n/a	B	516	ジェネレータON、 レジスタ無効を出力	off	512バイトの ユーザデータ+ヘッダーのための ECCを形成
Cmnd 2	0	B	Y	ジェネレータOFF、 レジスタ無効を出力	on	ユーザデータ1+ユーザデータ2 をフラッシュバッファに送信
Cmnd 3	続く	B + 512	4	ジェネレータOFF、 レジスタ無効を出力	on	ヘッダーを フラッシュバッファに送信
Cmnd 4	続く	n/a	12	ジェネレータOFF、 レジスタ有効を出力	on	ECCをフラッシュバッファに送信
Cmnd 5	続く	B + Y	512 - Y	ジェネレータOFF、 レジスタ無効を出力	on	ユーザデータ3を フラッシュバッファに送信し、 フラッシュバッファを フラッシュアレーに書き込む

【図 33】

読み込み操作終了後におけるコントローラのバッファメモリ内のデータ

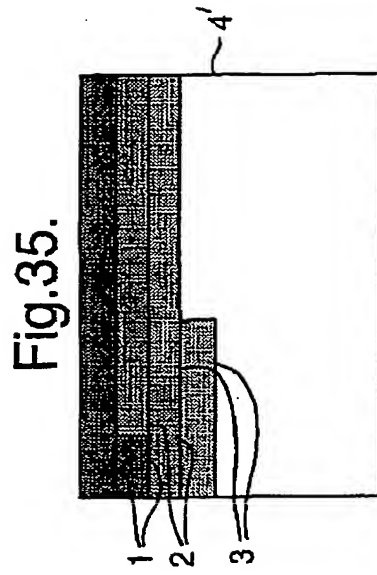


【図34】

コントローラのデータ送信ハードウェアへのコマンドシーケンス

コマンド シーケンス	コマンドパラメータ			コマンドの内容		
	フラッシュ 開始 アドレス	バッファ 開始 アドレス	データXfr サイクル の数	ECCジェネレータモード	フラッシュ クロック	概要
Cmnd 1	0	B	X	ジェネレータON、 レジスタ無効を出力	on	フラッシュページを フラッシュバッファに読み込み、 フラッシュバッファからの ユーザデータ1を送信
Cmnd 2	続く	B + X	Y - X	ジェネレータON、 レジスタ無効を出力	on	フラッシュバッファからの ユーザデータ2を送信
Cmnd 3	続く	B + 512	16	ジェネレータOFF、 レジスタ無効を出力	on	フラッシュバッファから ヘッダー+ECCを送信
Cmnd 4	続く	B + Y	512 - Y	ジェネレータON、 レジスタ無効を出力	on	ユーザデータ3を フラッシュバッファに送信
Cmnd 5	n/a	B + 512	16	ジェネレータON、 レジスタ無効を出力	off	ECCジェネレータから ヘッダー+ECCを送信

【図 35】



【国際調査報告】

INTERNATIONAL SEARCH REPORT

International Application No.
PCT/GB 00/00550

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 06F3/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	GB 2 291 991 A (MEMORY CORP PLC) 7 February 1996 (1996-02-07) page 5, line 15 -page 8, line 5; figures	1,2,4, 10,11, 15,17, 20,46, 47,52, 54-56
A	WO 97 37296 A (SINCLAIR ALAN WELSH ;MEMORY CORP PLC (GB)) 9 October 1997 (1997-10-09) page 3, line 21 -page 7, line 6; figures -/-	1,52, 54-56

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents:

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubt on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- "Z" document member of the same patent family

Date of the actual completion of the international search

25 May 2000

Date of mailing of the international search report

02/06/2000

Name and mailing address of the ISA
European Patent Office, P.B. 5018 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3018

Authorized officer

Moens, R

1

INTERNATIONAL SEARCH REPORT

International Application No.
PCT/GB 00/00550

C. (Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 712 067 A (MITSUBISHI ELECTRIC CORP) 15 May 1996 (1996-05-15) column 4, line 37 -column 6, line 32; figures	1-4, 10, 15, 17, 24, 46, 47, 52, 54-56
A	EP 0 522 780 A (IBM) 13 January 1993 (1993-01-13) claims 1, 9, 13	1, 52, 54-56
A	WO 94 20906 A (SYSTEMS LTD M ; SYSTEMS INC M (US)) 15 September 1994 (1994-09-15) page 2, line 21 -page 5, line 2	1, 48, 49, 52
A	US 5 602 987 A (HARARI ELIYAHOU ET AL) 11 February 1997 (1997-02-11) column 8, line 39 - line 67; figures 1A, 2	1, 13, 14, 16, 56, 57

INTERNATIONAL SEARCH REPORT

information on patent family members

International Application No

PCT/GB 00/00550

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
GB 2291991 A	07-02-1996	CN 1198226 A	04-11-1998
		EP 0852766 A	15-07-1998
		WO 9712325 A	03-04-1997
		JP 11511879 T	12-10-1999
WO 9737296 A	09-10-1997	EP 0896699 A	17-02-1999
EP 0712067 A	15-05-1996	JP 8137634 A	31-05-1996
		US 5905993 A	18-05-1999
EP 0522780 A	13-01-1993	JP 2582487 B	19-02-1997
		JP 5027924 A	05-02-1993
		DE 69223287 D	08-01-1998
		DE 69223287 T	28-05-1998
		US 5524230 A	04-06-1996
WO 9420906 A	15-09-1994	US 5404485 A	04-04-1995
		AU 6269994 A	26-09-1994
		CN 1098526 A	08-02-1995
		DE 69414556 D	17-12-1998
		DE 69414556 T	06-05-1999
		EP 0688450 A	27-12-1995
		FI 954235 A	08-11-1995
		IL 108766 A	05-12-1996
		JP 8510072 T	22-10-1996
		ZA 9401446 A	26-09-1994
US 5602987 A	11-02-1997	US 5297148 A	22-03-1994
		US 5535328 A	09-07-1996
		US 5991517 A	23-11-1999
		DE 69024086 D	25-01-1996
		DE 69024086 T	20-06-1996
		DE 69033262 D	30-09-1999
		DE 69033262 T	24-02-2000
		DE 69033438 D	02-03-2000
		EP 0392895 A	17-10-1990
		EP 0617363 A	28-09-1994
		EP 0618535 A	05-10-1994
		EP 0675502 A	04-10-1995
		EP 0935255 A	11-08-1999
		JP 2292798 A	04-12-1990
		US 5671229 A	23-09-1997
		US 5719808 A	17-02-1998
		US 5936971 A	10-08-1999
		US 5418752 A	23-05-1995
		US 5877986 A	02-03-1999
		US 5862080 A	19-01-1999
		US 5999446 A	07-12-1999

フロントページの続き

(72)発明者 オスペンスカヤ ナターリヤ ビクトロブ
ナ
ロシア連邦 194100、 サンクトペテルブ
ルク、 アイームリンスキー プロスペク
ト 29/20 アパト 63

(72)発明者 テイラー リチャード マイケル
イギリス イーエイチ22 2エイキュー
ミドロージャン、 ダルキース、 ニュー
ミルス ロード 41、 オールド ソーミ
ル ハウス

(72)発明者 ゴロベッツ セルゲイ アナトリービッチ
イギリス イーエイチ9 1エスイー エ
ジンバラ、 イースト メイフィールド
16、 1エフエル

Fターム(参考) 5B018 GA04 HA35 KA14 MA22 NA06
5B060 AB26
5B065 BA05 CC03 CC08